5-9-2012 12:00 AM

3D Velocity Retrieval and Storm Tracking Using Multiple Radars

Yong Zhang
The University of Western Ontario

Supervisor
John Barron
The University of Western Ontario Joint Supervisor
Robert Mercer
The University of Western Ontario

Graduate Program in Computer Science
A thesis submitted in partial fulfillment of the requirements for the degree in Doctor of Philosophy

Recommended Citation

Zhang, Yong, "3D Velocity Retrieval and Storm Tracking Using Multiple Radars" (2012). *Electronic Thesis and Dissertation Repository*. 536.
https://ir.lib.uwo.ca/etd/536

3D VELOCITY RETRIEVAL AND STORM TRACKING USING MULTIPLE RADARS

(Thesis format: Monograph)

by

Yong Zhang

Graduate Program in Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

THE UNIVERSITY OF WESTERN ONTARIO

School of Graduate and Postdoctoral Studies

CERTIFICATE OF EXAMINATION

Supervisor:	Examiners:
	....
....	Dr. Norman Donaldson
Dr. R. E. Mercer	
Supervisor:	....
	Dr. Wayne Hocking
	....
....	Dr. Sylvia Osborn
Dr. J. L. Barron	
	....
	Dr. Steven Beauchemin

The thesis by

Yong Zhang

entitled:

3D Velocity Retrieval and Storm Tracking Using Multiple Radars

is accepted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

....	....
Date	Chair of the Thesis Examination Board

Abstract

Severe weather forecasting is one of the most important and urgent tasks in the meteorology field. This thesis builds on previous work by Barron and Mercer and their graduate students, concerning the use of 3D optical flow to retrieve 3D wind velocity from 3D Doppler radial velocity datasets and tracking 3D severe weather storms using fuzzy points realized as ellipsoids to represent storms and a fuzzy algebra machinery in a relaxation labeling framework to track storms in Doppler precipitation datasets.

We first extend the original 3D optical flow (both least squares and regularization methods) for recovering 3D wind velocity from the multiple overlapping Doppler radial velocity fields. The enhanced methods exhibit improved performance, especially in overlapping radar areas. We also add 3D windprofiler data into our framework. We show that windprofiler data allows the vertical component of 3D velocity to be more accurately recovered. We perform a quantitative analysis on synthetic Doppler data and a qualitative analysis on real Great Lakes Doppler datasets and show that both multiple Doppler data and windprofiler data significantly improve the performance. Our optical flow general frameworks lends itself to adding new sources of data and new constraints on that data.

We also use a “pseudo” storm concept to solve the tracking problems caused by merging and splitting of severe weather storms over time. We first modify the original tracking algorithm to add a pseudo storm definition to it. Then, an advanced storm tracking algorithm taking full advantage of pseudo storms is presented. We compare the results using the original storm tracking algorithm, the original storm tracking algorithm with pseudo storms added and the final advanced pseudo storm tracking algorithm. The advanced pseudo storm tracking algorithm outperforms the other storm tracking algorithms for Great Lakes Doppler precipitation datasets.

Keywords: Doppler radar, windprofiler, 3D velocity retrieval, optical flow, least squares, regularization, 3D fuzzy point algebra, relaxation labeling algorithm, storm detection, storm tracking, pseudo storm, multiple radars.

Acknowledgements

I would like to thank my two thesis supervisors, Dr. John Barron and Dr. Robert Mercer for their incredible patience, tremendous help and insightful advice. It would have been impossible for me to accomplish this work without their guidance. Also, many thanks to Dr. Paul Joe for his tremendous help with the Doppler radar background, and Dr. Wayne Hocking for his detailed tutorial on the Windprofiler data.

Finally, I want to express my greatest appreciation to my parents for their unconditional support and endless love. They have been there for me throughout my entire academic career.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In meteorology, severe weather storms refer in part to the localized convection caused by updrafts and downdraughts in the air. Usually severe storms focus on a smaller area compared to tropical cyclone areas, with a duration varying from about half an hour to a couple of hours. Severe storms can be classified into single-cell, multi-cell or super-cell storms, according to their scale. They could produce violent weather phenomena in the form of thunderstorms, hail, heavy rains, or even tornadoes with heavy precipitation. Severe storms are the most common natural hazards and can cause significant critical lost of life and property damage. Therefore, an important task for meteorologists is to forecast the formation of severe storms. This task includes the detection of severe storms, the measurement of their sizes and motion and the tracking of storms through their life cycles.

1.1 Doppler Radar

Doppler radar has been considered to be a valuable observation tool in meteorology for a long time [59]. It is capable of observing high resolution information about the internal structure of severe weather storms hundreds of kilometers from the radar. In 1842, Christian Doppler [23] observed that sound waves would have a higher frequency if the source was moving toward the observer and a lower frequency if the source was moving away from the observer. This phenomenon not only applies to sound waves, but to all types of waves. If one measures how long it takes for a ordinary radar wave to reflect back from an environmental particle (say a rain drop) to the radar then one can use that time to calculate how far the particle is from the radar. The frequency it detects from a moving object can be calculated as in Equation 1.1:

$$f_{Doppler} = 2 * V_{target} \frac{f_{transmitted}}{c}, \tag{1.1}$$

where V_{target} is the velocity of the object according to the radar, $f_{transmitted}$ is the frequency originally transmitted from the radar and c is the velocity of light.

With a Doppler radar, one can not only compute whether a raindrop is moving toward or away from the radar but the speed of the rain drop as well (along the line from the radar to the rain drop). This calculation is based on the rate of compression or expansion of the radar wave, i.e. the Doppler effect. The speed and radial direction together comprise the radial velocity. Since wind causes rain drops to move, this radial velocity is actually a type of wind velocity measurement. Precipitation density (reflectivity) relates to the amount of the rain in a unit volume and is measured by the strength of the reflected wave.

Figure 1.1 depicts the reflectivity image detected by Doppler radar in Sept. 16^{th} 1999, 10:50. The different colours in the image represent the different densities of reflectivity, according to the colour map. As the colour changes, the magnitude of reflectivity could be up to 65 dBZ (decibels of Z). If the reflectivity is $>$ 65 dBZ the rain is extremely heavy, between $46 - 65$ dBZ it is heavy, between $24 - 45$ dBZ it is moderate, between $8 - 23$ it is light and between $0 - 8$ dBZ there is barely any rain at all[1].

The actual reflectivity values detected by radar are at discrete voxel locations. Fig 1.1 is the result of a bilinear patch algorithm [56] that was used to smooth/fill in this voxel data. Without bilinear interpolation the image would looks like a sparse collection of rays transmitting from the radar center. Later we will see the recognition and tracking of severe storms are all based on this reflectivity image [4, 5, 17, 56, 57, 68, 69, 70].

---

[1] http://en.wikipedia.org/wiki/DBZ_(meteorology)

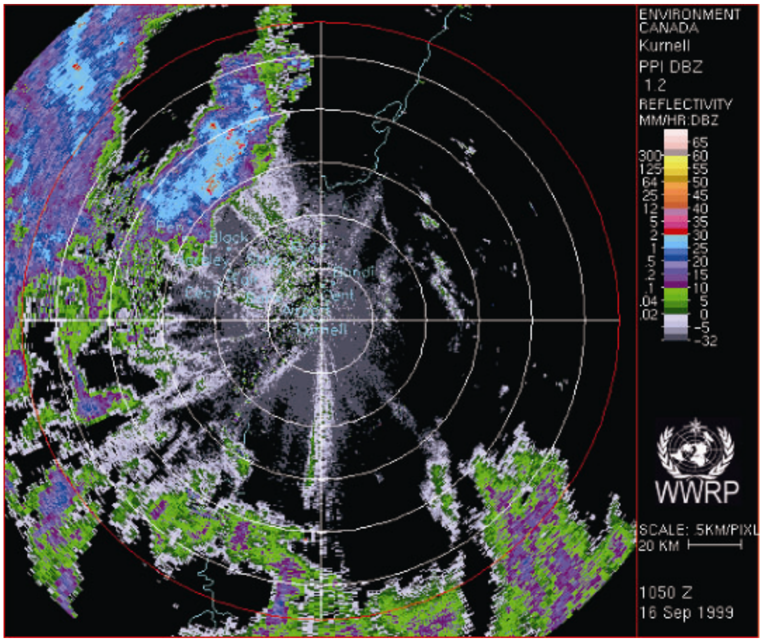


Figure 1.1: An Example of Doppler Radar Reflectivity Image(1999, September 16th, 10:50) provided by Dr. Paul Joe from Environment Canada

Figure 1.2 shows the radial velocity image automatically generated by radar. It was acquired at the same time as the reflectivity image in Figure 1.1. The coloured parts cover a similar area as in the reflectivity case, demonstrating the wind movements over that area. The different colours here represent not only the magnitudes but also their directions, as shown in the colour map. Basically, "Red" represents a positive direction, which means the wind is moving away from the radar, while "Blue" implies a negative wind velocity, which is moving towards the radar. It must be noted that the velocity detected by radar is only the radial parts (i.e. in the direction of transmitting beams). Earlier work has shown how to utilize this knowledge to retrieve an approximation to the full 3D velocity of wind field [6, 13, 14, 15].

The coordinate system used in the 3D velocity representation is shown in Figure 1.3a. The x axis is left to right, the y axis is from bottom to top and the z axis is height (upwards). The origin is at (0,0,0). 3D velocity $\vec{V}$ has 3 components (U, V, W).

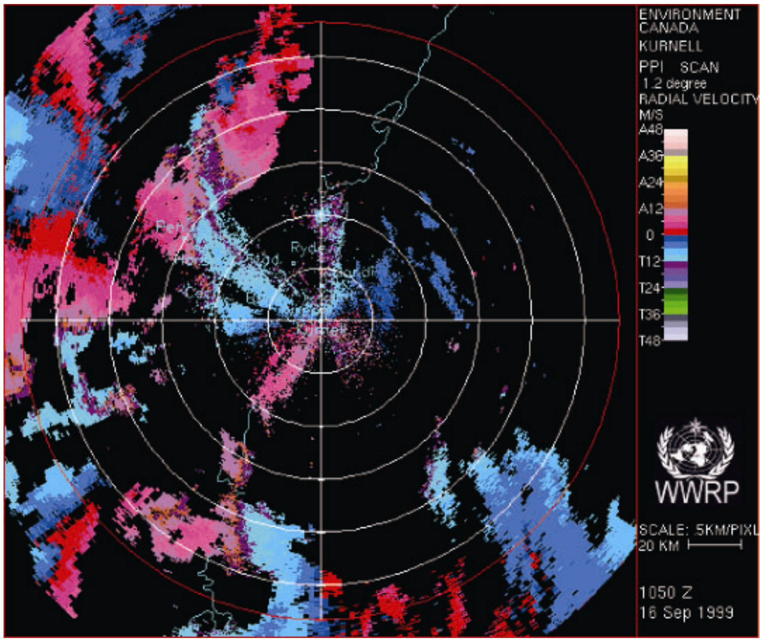
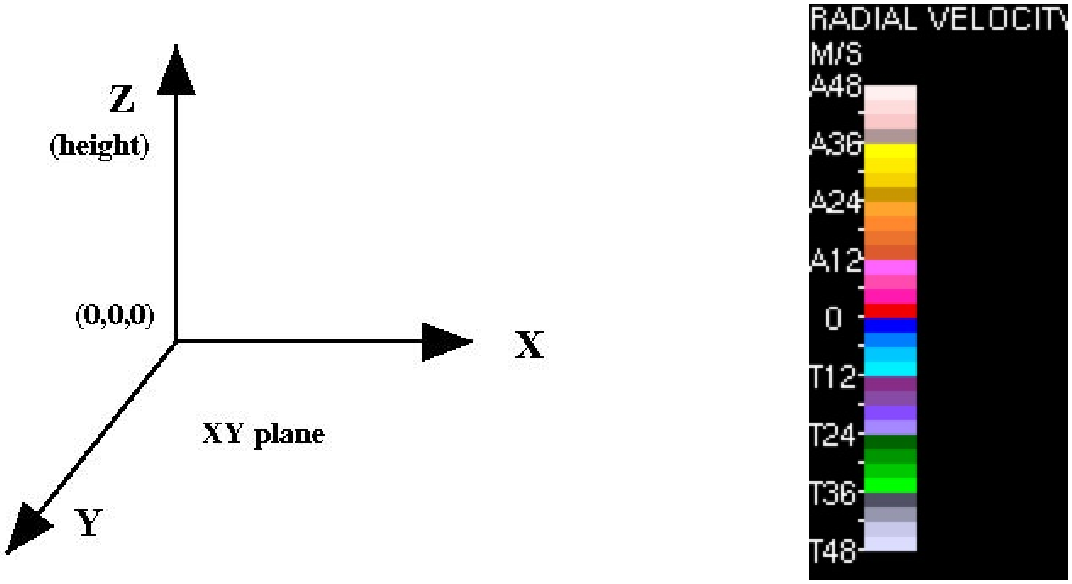


Figure 1.2: An Example of Doppler Radar Velocity Image(1999, September 16th, 10:50) provided by Dr. Paul Joe from Environment Canada

Figure 1.4 illustrates the 3D structure of NEXRAD *I* (**Nex**t-Generation **Rad**ar Level I) Doppler radar data. The radar data covers approximately a circular cone, with area roughly about $87,000\ km^2$. There are 15 elevations of data with the cone angle centered at the radar location, changing from the minimum angle Φ_{min} of 58° to a maximum angle ϕ_{max} of 89.5°. The length of cone radii varies from a minimum 508.84 *km* to an maximum radius of 599.97 *km*. Vertically the height of elevation is from the ground level, 5.25 *km*, up to 317.4 *km*. At each elevation, there are 360 beams transmitting (one beam per degree of a circle). For each degree of the beam, there are 600 points where reflectivity/radial velocity data is measured. These points are referred to as voxels because they represent the integration of such data in 3D frustums at those points. Table 1.1 shows more details about the angle (measured according to the vertical axis). All previous research uses this type of radar data [4, 5, 17, 56, 57, 68, 69, 70].

In the past decades, the NOAA (National Oceanic and Atmospheric Administration)



(a) (b)

Figure 1.3: (a) The coordinate system used by 3D velocity representation and (b) the colour-magnitude correspondence map used for Doppler radial velocity.

has placed several Doppler radars around the Great lakes area. They are using the next generation of NEXRAD radar data called NEXRAD *II*. The NEXRAD *II* datasets share a similar structure with the NEXRAD *I*. However, its data structure is capable of storing dynamic parameters, so that the number of elevations and resolution of the radar beams can change. Generally, a NEXRAD *II* radar has fewer elevation numbers than a NEXRAD *I* radar, about only 9 elevations down from 15 before (Usually the higher elevations contain fewer data so they are not that useful). There are 920 voxels to record reflectivity data, covering a circular area with radius of 460 *km* at each elevation. So the coverage area of reflectivity data is smaller but the resolution along each beam is twice as high as before. There are 920 voxels to record radial velocity data, but the radius of the coverage circle is only 230 km. Thus the radial velocity data covers a much smaller area (half of the reflectivity data area and about a quarter of NEXRAD *I* data area) with a resolution of 0.25 *km*. This modification has as a significant impact on the calculation of the upwards velocity vector since there is only smaller variation of the data in this dimension. Table 1.1 shows the elevation angles for the NEXRAD *II* data. It can be seen that the overall number of angles for the NEXRAD *II* data are smaller than the NEXRAD *I* angles. The NEXRAD *II* data focuses more on the lower height area than NEXRAD *I*, where the *W* velocity component is almost orthogonal to the radial velocities. So radial velocity contains little reliable *W* information (and is extremely sensitive to noise). This is an example of the Aperture Problem [3], where most of the local velocity information is orthogonal to underlying 3D velocity. We will show how to overcome or attenuate this problem using algorithms presented in this thesis.

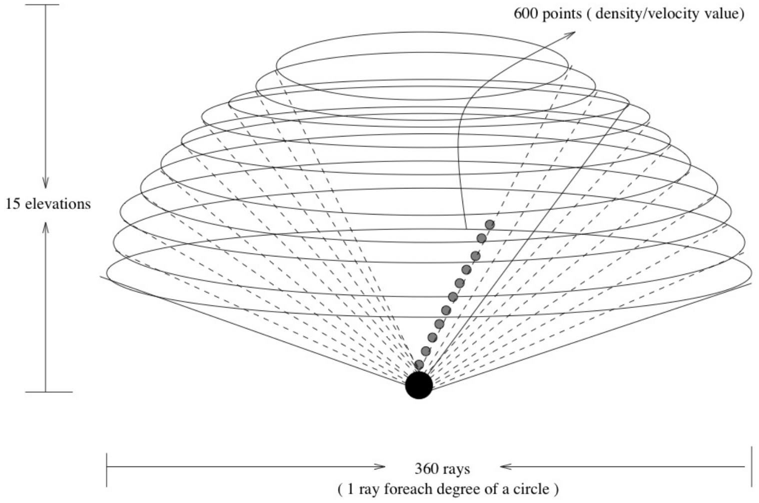


Figure 1.4: The structure of NEXRADI Doppler radar dataset. The NEXRADII data is structured in a similar way.

1.2 Our Problems

In order to understand how storms develop and move over time, much research has been devoted to retrieving 3D full wind velocity from the observed radial velocity (for example, see Lhermitte and Atlas [45], Easterbrook [27] and Waldteufel and Corbin [72]). Rather

Elevation Number	NEXRAD *I* Angle (ϕ°)	NEXRAD *II* Angle (ϕ°)
0	89.5	89.5
1	88.8	88.51
2	88.1	87.5
3	87.3	86.57
4	86.5	85.65
5	85.3	83.94
6	84.0	80.07
7	82.5	75.37
8	80.8	70.44
9	79.0	NA
10	77.0	NA
11	74.0	NA
12	70.0	NA
13	65.0	NA
14	58.0	NA

Table 1.1: The elevation angles (angles of the cone walls with the positive z axis) of NEXRAD *I* and NEXRAD *II* Doppler radars

than using the traditional methods provided by meteorologists (see Chapter 2), we solve this problem using the 3D Optical Flow framework ([14, 15, 69]), which is a technology widely applied in the Computer Vision area. 2D optical flow estimates the 2D image motion of pixels in an image taken from an image sequence. The 3D extension of 2D optical flow allows the computation of the 3D volumetric motion of voxels in a sequence of 3D volumes. In meteorology applications, 3D optical flow is a measure of 3D wind velocity. Reliable performance has been obtained on real 3D Doppler radar data (NEXRAD *I*) [14, 15] using both the least squares [48] and a global regularization [37] optical flow frameworks.

We used 3D optical flow to recover full 3D wind velocity from radar data measured by NCDC Doppler radars in the Great Lakes area. However, previous optical flow work [4, 5, 17, 56, 57, 68, 69, 70] used only one Doppler radar. In the Great Lakes region, the coverage areas of the radars often overlap and this suggests the possibility to combine them. This thesis shows how to use multiple Doppler radars in both the Least Squares and the Regularization frameworks to enhance the performance. We call the advanced least squares method the **Dual LS(Least Squares)** approach as opposed to the original Single LS approach proposed before [14], and refer to our refined regularization method as the **Dual Regularization** approach as opposed to the original Single Regularization approach proposed before [15]. In addition to qualitatively evaluating the optical flow results for the real Great Lakes radar datasets, we also quantitatively examine the optical flow performance using various synthetic Doppler radar datasets.

Recently, another type of Doppler radar, the Ontario-Quebec VHF Windprofiler Radar Network (O-Q net) has also been installed around the same Great Lakes area. These radars are also Doppler radars, but compared to the traditional precipitation-based Doppler radars as we previously discussed, they cover a relatively smaller area and do not have the limitations in the upward direction for wind velocity retrieval. The windprofiler radar uses radio waves to detect the wind speed and direction at various elevations above the ground and work even when little or no precipitation is not present. It is believed that windprofiler radars can provide accurate local wind measurements up to 15 *km* high. Therefore in this thesis, we consider how to integrate the data from windprofiler to precipitation-based Doppler radar, which allows more accurate wind measurements in the overlapping radar areas, especially in the upwards direction (where Doppler radar wind recovery is weak). We present another modification to the regularization method used above, which we call the “refined” regularization approach, to integrate windprofiler and Doppler data. Again, we qualitatively evaluate our flow on real Great Lakes windprofiler and Doppler datasets and quantitatively on synthetic windprofiler and Doppler datasets.

After showing how full 3D wind velocity can be recovered using radial velocity, we investigate storm tracking using the reflectivity data provided by Doppler radar (3D velocity is one compatibility function in our tracking algorithm). Based on previous work [56, 68, 69, 70] we propose a new storm detection and tracking algorithm that builds on previous work and is capable of working in a multiple-Doppler radar environment. To handle the complicated storm patterns resulting from using multiple overlapping radars, we expand on a novel concept called "Pseudo Storms". This idea was initially introduced by Krezeski et al. [41] for 2D storm tracking. In addition to 3D pseudo storm, we present a tracking algorithm that uses this concept to track storms that not only change their shapes and sizes over time but also merge with other storms to a bigger storm or split into a number of smaller storms. Krezeski et al. did not present a pseudo storm tracking algorithm [41]. A comparison between the original tracking algorithm (see [5, 17, 70]) and the new pseudo tracking algorithm is given.

Figure 1.5 shows the distribution of Doppler radars and other radars such as windprofilers around the lower Great Lakes area in North America, especially around Lake Erie. We have acquired NEXRAD *II* data from the Detroit and Cleveland radars via the NCDC (National Climate Data Center) network in the US. We have also acquired Canadian Doppler radar data from the King City and Exeter radars.

1.3 Thesis Contributions

We briefly enumerate the contributions of this thesis:

1. We propose a generalized framework to compute 3D optical flow in order to recover the 3D full wind velocity via multiple Doppler radars.

2. We have integrated windprofiler radar data into our framework, to enhance the accuracy of wind velocity recovery, especially in the upward direction.

3. Experiments using both real data and synthetic data are used to qualitatively and quantitatively evaluate the performance of our algorithms.

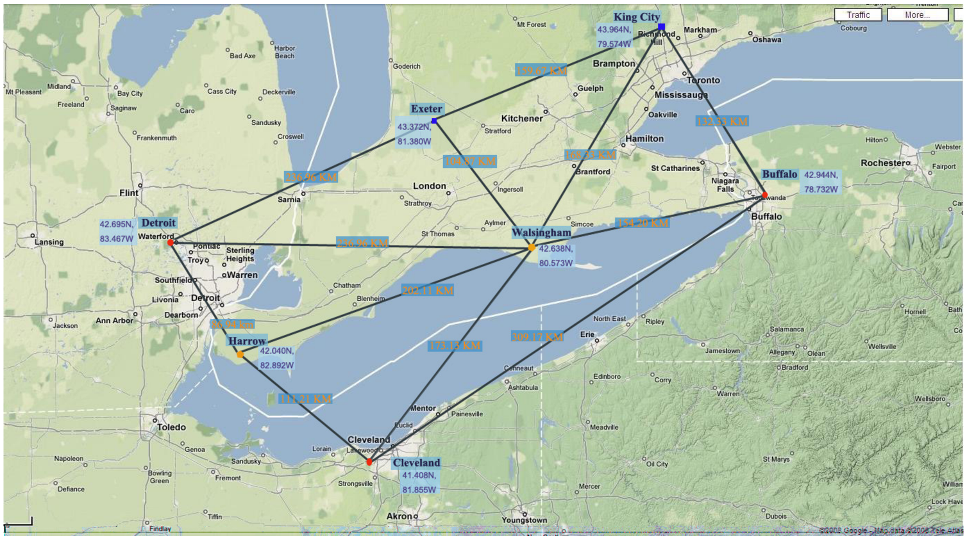


Figure 1.5: The Doppler and windprofiler radars in southwestern Ontario and the adjacent Northern American area states (the Great Lakes area). The Doppler radars at Detroit, Cleveland and Buffalo are shown as red dots, the Doppler radars at King City and Exeter are shown as blue dots and the windprofiler radars are at Harrow and Washingham are shown as orange dots.

4. We introduce the pseudo storm idea as a better representation of the continually deforming real storms. We have redesigned the original tracking algorithm to implement pseudo storm into it.

5. We propose an advanced storm tracking algorithm using pseudo storm with its full advantages. This new pseudo storm tracking solution can generate reasonable results in a more comprehensive situation, compared to the original tracking algorithm.

1.4 Overview of Thesis

Chapter 2 presents a literature survey of the previous work on 3D wind recovery from single and multiple radars. First, we review velocity retrieval using Doppler radars, then using multiple (dual) Doppler radars and finally using windprofiler radars. The second part of the literature survey deals with storm detection and tracking in Doppler radar datasets.

After the literature survey, the thesis is divided into two parts. In PART I, we concentrate on the recovery of full 3D wind velocity using optical flow methods from single and multiple Doppler and/or windprofiler radars. In PART II, we show how to detect and track severe storm in Doppler radar datasets. Chapters 3, 4, 5 belong to PART I. Chapters 6, 7 belong to PART II.

Chapter 3 presents the modified least squares and regularization methods used to recover 3D wind velocity using single or multiple Doppler radar datasets.

Chapter 4 presents the design and quantitative analysis of our synthetic Doppler radar datasets. The results for the Great Lakes real Doppler radar datasets are also presented.

Chapter 5 shows the integration of windprofiler radar data into our optical flow solution, adding extra information on the upward direction of 3D wind velocity to our optical flow algorithms. We quantitatively and qualitatively evaluate our algorithm on synthetic and real Doppler and windprofiler datasets.

Chapter 6 presents the storm detection algorithm in Doppler radar datasets and the data structures necessary to represent them. To solve the limitations of the original method, we introduce a “pseudo storm” idea. We also modify the original tracking algorithm to use a restricted pseudo storms definition in our program, which shows advantages but still has its limitations.

Chapter 7 presents the advanced storm tracking algorithm using pseudo storms thoroughly to generate better tracks than both the original tracking algorithm and the modified one (that only has restricted pseudo storms implemented). Results from all the algorithms are displayed and compared.

Finally, Chapter 8 gives our conclusions and the suggestions for future work.

Chapter 2

Literature Survey

The original data obtained from precipitation-based Doppler radar comprise:

1. the reflectivity or precipitation density,
2. the radial component of full wind velocity in the direction of radar beam and
3. the spectrum width (effectively the spectral and temporal variance in the local radial velocities).

Reflectivity and radial velocities are used for full velocity retrieval and storm tracking, while the spectrum width (effectively, the variance of the radial velocities within a voxel) can help estimate the reliability of radial velocity. We use a spectrum width threshold of 30 dBZ to identify unreliable radial velocities in this thesis.

To better understand the motion of wind in the atmosphere (and, hence, the motion of severe storms as they move with the wind), meteorologists need more detailed 3D information, such as the 3D wind velocity field. Since the very beginning of radar application for severe storm nowcasting/forecasting, much research has been devoted to the use of Doppler radar. The initial application in the early days can be found in R. J. Doviak and D. S. Zrnić's book [25]. Computer Vision techniques allow radar data to be treated as images of storm motion and deformation. By analyzing the attributes of these images, more detailed and accurate views of 3D wind motion can be obtained. Based on this information, a more reliable prediction of severe storms behavior can be predicted.

2.1 Retrieving 3D Velocity by Single Doppler Radar

A significant amount of research has been devoted to the retrieval of 3D full wind velocity. Since the Doppler radar can only measure the radial wind velocity, additional constraints must be employed to recover 3D velocity. In order to do that, one hypothesis made was that there is only very little modification of wind field when it moves across the radar's view range [45]. As well, all the data in a radar's data volume is assumed to be detected simultaneously (actually there is a significant time difference between the times the first and last data items are measured due to the radar rotating 360° for each elevation of data acquired). In the following sections below we present traditional approaches to the recovery of 3D full wind velocities using data from a single Doppler radar. We also discuss the use of optical flow techniques from the Computer Vision area to measure 3D wind velocity (we use the words **velocity** and **flow** interchangeably). The traditional approaches try to fit the observed radial velocities into various models while the optical flow approach uses various constraint of 3D motion, which are not directly derived from the physics of wind motion.

2.1.1 Traditional Approaches to Retrieve 3D Wind Velocity

In order to simplify the retrieval process in the single Doppler radar case, usually an assumption is made that wind properties changes **linearly**. Lhermitte and Atlas [45] proposed the VAD (Velocity Azimuth Display) approach to estimate the mean horizontal wind magnitude (in the 2D plane) and the direction about horizontal circles centered about the vertical axis of radar. It was shown that some parameters of the wind velocity, such as the mean convergence, divergence and deformation, could also be retrieved by this method. Furthermore, Easterbrook [27] proposed that this retrieval could also be performed in a conical sector, called VARD (Velocity ARea Display), which usually provides more reliable results with the lower elevations in the clean air case. Later, Waldteufel and Corbin [72] extended VAD and VARD into a more sophisticated technique, the VVP (Volume Velocity Processing) technique, which processes radar data in a volume. This solution is the most advanced one among all the traditional solutions.

2.1.1.1 The VAD Analysis Procedure

For any given environmental point (x_0, y_0, z_0), the linearity assumption for wind field $\vec{V} = (u, v, w)$ can be expressed as:

$$\begin{aligned} u &= u_0 + u'_x(x - x_0) + u'_y(y - y_0) + u'_z(z - z_0) \\ v &= v_0 + v'_x(x - x_0) + v'_y(y - y_0) + v'_z(z - z_0) \\ w &= w_0 + w'_x(x - x_0) + w'_y(y - y_0) + w'_z(z - z_0) \end{aligned} \tag{2.1}$$

where $\vec{V} = (u, v, w)$ is the 3D wind velocity that varies linearly around its value (u_0, v_0, w_0) at a point (x_0, y_0, z_0). The u' and so on are the derivatives of velocity components on the x, y, z axes.

At the same time, the magnitude of the radial velocity can be estimated in Polar coordinates as:

$$V_r = u \cos\theta \cos\phi + v \sin\theta \cos\phi + w \sin\phi \tag{2.2}$$

where θ is the azimuth angle while ϕ is the elevation angle. Combing these two equations and using R as the radial distance from an environmental point to the radar, Lhermitte and

Atlas [45] obtain:

$$
\begin{aligned}
\mathbf{V_r} &= \cos\theta\cos\phi(u_0 - u'_x x_0 - u'_y y_0 - u' z_0) \\
&+ \sin\theta\cos\phi(v_0 - v'_x x_0 - v'_y y_0 - v' z_0) \\
&+ \sin\phi(w_0 - w'_x x_0 - -w'_y y_0 - w'_z z_0) \\
&+ R\cos^2\theta\cos^2\phi u'_x \\
&+ R\sin^2\theta\cos^2\phi v'_y \\
&+ R\sin\theta\cos\theta\cos^2\phi(u'_y + v'_x) \\
&+ R\sin^2\phi\omega'_z \\
&+ R\cos\theta\sin\phi\cos\phi(u'_z + w'_x) \\
&+ R\sin\theta\sin\phi\cos\phi(v'_z + w'_x)
\end{aligned}
\tag{2.3}
$$

In order to solve this equation, Lhermitte and Atlas neglect some parameters such as vertical derivatives w'_x, w'_y or assume they are comparatively small. In order to retrieve u_0 and v_0 without further information, they choose $x_0 = y_0 = 0$. In this way, a 2D linear least squares regression procedure was applied to solve this system of linear equations. In this VAD approach, only terms with different azimuth dependence can be discriminated. However, the noises from vertical velocity term can seriously contaminate the final results, so its application is limited to only the low elevations. Their method is similar to that of Lucas and Kanade optical flow [48], but in contrast they assume a global constant velocity rather than Lucas and Kanade's local constant velocity assumption.

Caya and Zawadzki [11] examined the performance of VAD in nonlinear wind fields. Their work showed that the parameters retrieved by a VAD analysis have no clear physical meaning if the linearity assumption cannot be guaranteed. In order to improve the reliability of this retrieval process, some refinements, such as VARD [27], EVAD [20] and VVP (Volume Velocity Processing) [72], have been proposed.

2.1.1.2 The VVP Analysis Procedure

Easterbrook proposed a method that discriminates every radial velocity term in Equation 2.4 on a conical surface (which each elevation of Doppler data is) by varying θ and R. The overall results are satisfying but might have a negative effect on the vertical velocity components and their divergence due to possible large-scale irregularities in the real data. Waldteufel and Corbin in [72] generalized Easterbrook's approach to apply to a series of conical scans (multiple elevations of Doppler data). If the total acquisition time is short enough to guarantee immediate processing of all the data, then the divergence and vertical velocity terms can be discriminated. They claim that this VVP (Volume Velocity Processing) method could not only retrieve the horizontal mean velocity and their derivatives without any contamination, but also provide a rough estimation of the vertical wind velocity components. Waldteufel and Corbin [72] performed an error analysis for their method on both the simulated and real data and found that the accuracy of 3D velocity estimates are limited by the inadequacy of the linear velocity assumption. Furthermore, they accommodated the effect of nonlinearities on the recovered wind field by introducing higher order terms as the ones in Equation (2.4). They found the nonlinear fluctuations (due to violations of the linear assumption) can be averaged to a large extent and they obtained reliable results in the higher elevations compared to VAD, except for negative effects due to the vertical velocity w and its derivative w_z'.

Boccipio [9] claims the primary difference between the VAD and VVP approaches is the way in which parameter-basis function decomposition is chosen from Equation (2.4): VAD and its extensions choose adequately sampled functions first while the VVP method selects the desired parameters and then determines the basic functions accordingly. Boccippio diagnosed the retrieval carefully with different regression methods under both clear air and heavy precipitation circumstances. They found the VVP method is able to recover a detailed velocity field, especially in clear air situations (where no obvious precipitation can be observed). The potential bias to real velocity could be estimated by direct diagnostics such as *RMSE* (root mean square error) and *Condition Number* examinations. Koscielny *et* *al.* [40] applied this VVP technique successfully in real pre-storm boundary-layer observation and showed that their results were verified via a dual-Doppler radar observation.

The VVP approach seems to be more powerful when dealing with the higher elevation data than the VAD method. It can process large quantities of data simultaneously. It is theoretically better at retrieving the vertical component w and its derivative $w_z^{'}$ but is still very sensitive to the linear irregularities of wind field [9, 72].

2.1.1.3 Further Developments

All these traditional methods discussed above are based on the assumptions that the wind field is smooth and changing linearly, so their applications have been limited to mostly the clear air case. In order to retrieve variational wind velocities, Qiu and Xu [55] have proposed a simple adjoint method, which can obtain the horizontal component of the wind field using the conservation of reflectivity and the momentum equation of radial velocity as physical constraints. Laroche and Zawadzki [43, 44] suggested a similar method but applied a prognostic function in its Lagrangian format, imposing the reflectivity conservation as a weak constraint instead. Shapiro *et al.* [61] presented a new 3D single-Doppler velocity retrieval method, which involves the usage of temporal constraints on the velocity field to smooth the data over time. They present results for two radars on micro-burst data. Based on this work, Shapiro *et al.* [63] combined aspects of Qiu and Xu's adjoint retrieval [55] and Laroche and Zawadzki's Lagrangian methods [43, 44] and proposed a novel approximate dynamical retrieval process of wind field using data from a single Doppler radar. Their method was tested on real data with a VVP estimation of the wind field as a background constraint.

More recently, Gong and Xu [32, 76, 78] have focused on how to use the VAD method to solve the aliasing problem in Doppler Radar's radial velocity data. They first proposed a three-step dealiasing method [32] to control the data quality, which requires additional processing of the radar velocity data before and after applying the traditional VAD method, such as the preliminary reference check to filter out noises in raw data and continuity check to reduce noisy in smaller ranges. An updated VAD solution [79] was presented using a cost function to minimize the raw aliased radial velocity data. This method's application is limited to the uncertainty of global minimization of the whole dataset due to the existence of multiple local minima. In order to fix this problem, Xu [77] proposed a high-efficient two-step VAD algorithm to find the global minimization after adopting a proper transformed subspace. Xu et al. [78] proposed a more simplified yet more accurate VAD-based solution to the radial velocity dealiasing problem, that they called alias-robust VAD analysis. The performance of this method was compared with the traditional VAD methods using radar data under various weather conditions. They concluded that the new AR-VAD (Alias-Robust VAD) method surpasses the original ones in eliminating and avoiding false dealiasing. However, its application still requires the VAD uniform-wind assumption.

It should be noted that the reliability of these wind velocity measurements under extreme weather conditions remains unclear because then the reflectivity conservation constraint is violated there. In general, the reflectivity conservation constraint is equivalent to the motion constraint equation below, except that it is given in spherical polar and Cartesian coordinates and usually includes additional terms, such as the rate of rain fall.

2.1.2 3D Optical Flow Solution to Retrieve 3D Velocity

Also known as the "Image Velocity" problem, the traditional 2D optical flow is an approximation of image motion for each pixel in an image. The computed flow is often used to retrieve the 3D motion and structure parameters of camera scene, to detect object motion and image segmentation. Therefore the estimation of optical flow has been widely applied in the computer vision area [39] to provide essential cues for the 2D and 3D motion and scene analysis.

There are many solutions proposed to estimate the optical flow field, which can be roughly classified into three categories: *differential methods*, *frequency-based methods*, and *matching methods*. The boundaries between these classes often overlap with each other and the differential method is usually involved in some way in most algorithms. Lucas and

Kanade [48] and Horn and Schunck [37] proposed two differential optical flow methods (now considered as classical). Lucas and Kanade use local least squares calculations while Horn and Schunck globally regularize the computed flow to minimize an energy function. These and other optical flow approaches have been subjected to both a qualitative and a quantitative analysis [3].

2.1.2.1 Traditional 2D Optical Flow

In the differential case, the calculation of optical flow is based on the 1^{st} order spatial-temporal intensity derivatives at each image pixel. There are many approaches to solve the optical flow problem. Most differential methods assume that local neighborhood intensity structure of local moving intensity regions remain the same between adjacent images. That is, they assume local translation (no rotation) and rigid body motion (no deformations in the objects) which generally holds if the time interval between adjacent images is small. It is also assumed that local intensity changes are due entirely to object/camera motion. The degree to which these assumptions are true determines the reliability of the measured image velocity. Based on this assumption, it is east to derive the **motion constraint** or **optical flow constraint** equation:

$$I_x u + I_y v + I_t = 0. \tag{2.4}$$

This is one equation with two unknowns, the x and y components of image velocity (u, v). I_x, I_y and I_t are measured from the image and is preceding and following images using convolution of a differentiation filter with the image data at each pixel. Since the correct (u, v) are one point on the line, additional constraints are needed to solve for them uniquely. This line is a manifestation of the "Aperture Problem", where locally only velocity normal to the intensity structure, the so called **normal velocity** can be recovered as the (u, v) point on the line closest to the origin. Note that the normal velocity, $\overrightarrow{v_n} = (u_n, v_n)$ can be expressed in terms of the spatial-temporal 1^{st} order derivatives:

$$\overrightarrow{v_n} = \frac{-(I_x, I_y)I_t}{||(I_x, I_y)||_2^2}. \tag{2.5}$$

One additional constraint that can be used to recover full velocity $\overrightarrow{v} = (u, v)$ is integrating all local estimates of normal velocity into a full velocity. The Lucas and Kanade method uses a local least squares calculation to integrate the (usually) different normal velocities in some neighbourhood Ω centered at a pixel into a full velocity at that pixel. It is easy to see that if two neighbouring pixels have different normal velocities then the 2 motion constraint lines intersect at a single unique velocity. When multiple normal velocities are available then the least squares framework find the "best" intersection point (image velocity).

For a $n \times n = N$ neighborhood, Lucas and Kanade solve the following linear system of equations in the least squares framework:

$$\begin{aligned} I_{x0}u + I_{y0}v &= -I_{t0}, \\ I_{x1}u + I_{y1}v &= -I_{t1}, \\ I_{x2}u + I_{y2}v &= -I_{t2}, \\ &\vdots \\ I_{xN}u + I_{yN}v &= -I_{tN}. \end{aligned} \tag{2.6}$$

which can be rewritten as $A_{n\times 2}\overrightarrow{v} = B_{n\times 1}$. The full velocity then can be calculated by solving a 2×2 linear system of equations as:

$$\overrightarrow{v} = (A^T W^2 A)^{-1} A^T W^2 B, \tag{2.7}$$

where W is a diagonal matrix where the diagonal elements are weights that usually emphasize data closer to the central pixel than pixels further away. It is common practice to use values of a 2D Gaussian as their values [3]. The quality of $\overrightarrow{v}$ can be determined by the condition number κ of the integration matrix $A^T W^2 A$ or the eigenvalues, $\lambda_1 \leq \lambda_2$ and unit eigenvectors $\hat{e}_1$ and $\hat{e}_2$. Barron *et al.* [3] used a threshold on the smallest eigenvalue $\lambda_1 > 1.0$ in their analysis.

Horn and Schunck [37] resolved the aperture problem by adding a global smoothness constraint that assures that the full velocity vectors in the neighborhood vary smoothly. They combine this smoothness constraint with the motion constraint into an cost or energy term that minimizes over the entire flow field. Equation (2.8) shows the constraint (functional), F, that they minimize (or regularize) everywhere in the image:

$$F = \int\int (I_x u + I_y v + I_t)^2 + \alpha^2(u_x^2 + u_y^2 + v_x^2 + v_y^2)\partial x \partial y. \tag{2.8}$$

The Euler-Lagrange equations can be used to minimize this energy function, and derives the iterative solutions of $\overrightarrow{v}$ using the Gauss-Seidel method as:

$$\begin{aligned} u^{n+1} &= \overline{u}^n - \frac{I_x(I_x\overline{u}^n + I_y\overline{y}^n + I_t)}{\alpha^2 + I_x^2 + I_y^2 + I_t^2} \quad \text{and} && (2.9) \\ v^{n+1} &= \overline{v}^n - \frac{I_x(I_x\overline{u}^n + I_y\overline{y}^n + I_t)}{\alpha^2 + I_x^2 + I_y^2 + I_t^2} && (2.10) \end{aligned}$$

A new set of velocity estimates (u^{n+1}, v^{n+1}) is computed from estimated derivatives I_x, I_y, I_t and the previous average velocity estimates $(\overline{u}^n, \overline{v}^n)$ at each pixel. Iteration is stopped when a pre-set number of iterations have been performed, or the norm of the difference between the 2 adjacent optical flow fields is less than the required threshold.

2.1.2.2 3D Optical Flow

The standard optical flow constraint equation can easily be extended to the 3D case. Chaudhury *et al.* [12] formulated a 3D optical flow constraint using I_x, I_y, I_z, I_t derivatives. In this case, using the spatial-temporal derivatives, I_x, I_y, I_z and I_t, we can compute the normal velocity the same as for the 2D case as:

$$\vec{V}_n = \frac{-I_t(I_x, I_y, I_z)}{||\nabla I||_2^2}. \tag{2.11}$$

In this equation ∇I is the spatial intensity gradient, (I_x, I_y, I_z). The 3D normal velocity is perpendicular to the local intensity structure as before, but depending on the local structure, the aperture problem manifests itself as **plane normal** velocity (the local intensity structure fits a plane well) and **line normal** velocity (the local intensity structure is the intersection of 2 planes at a line). In the event that 3 or more planes can be used to describe the local intensity structure a full 3D velocity can be recovered. 3D normal velocity is described in more detail elsewhere [7, 64, 65].

Doppler radars supply reflectivity (precipitation density) data. However, because severe weather storms are deformable objects and there is no obvious relationship between storm motion and reflectivity changes, a differentiable optical approach to optical flow will probably fail here (see Laroche *et al.* [43, 44]). Instead, we use the 3D radial velocity directly in the 3D motion constraint equation to compute optical flow. That is, we use radial velocity as we would planar normal velocity in our algorithms. This means we do not concern ourselves with Doppler intensity derivative calculation at all. Radial velocity is directly measured by Doppler radar and is the sole input to our optical flow algorithms.

The 3D optical flow constraint equation:

$$I_x U + I_y V + I_z W + I_t = 0, \tag{2.12}$$

now can be rewritten as:

$$V_n = \vec{V} \cdot \hat{n}, \tag{2.13}$$

where:

$$\vec{V} = \frac{(I_x, I_y, I_z)}{\|(I_x, I_y, I_z)\|_2} \tag{2.14}$$

and

$$\hat{n} = \frac{-I_t}{\|(I_x, I_y, I_z)\|_2}. \tag{2.15}$$

In the above equations, $\vec{V} = (U, V, W)$. Note that $\vec{V}_n$ is the velocity with the smallest magnitude that satisfies Equation (2.12). The line from that point to the origin is perpendicular or normal to the optical flow line (and, hence, the name normal).

The 2D Lucas and Kanade optical flow method can be easily extended into 3D as:

$$\begin{pmatrix} n_{x1} & n_{y1} & n_{z1} \\ n_{x2} & n_{y2} & n_{z2} \\ & \vdots & \\ n_{x1} & n_{y1} & n_{z1} \end{pmatrix} \begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} V_{n1} \\ V_{n2} \\ \vdots \\ V_{n3} \end{pmatrix}. \tag{2.16}$$

where A and B are the $n \times 3$ and $n \times 1$ matrices that one should expect. Usually these systems of equations are set up for 3D neighbourhoods of normal velocities centered at the voxel for which we are computing the full velocity. Note that we have used the normal velocity magnitudes and direction components in our equations instead of the spatial-temporal derivatives to obtain the linear system of equations $A\vec{V} = B$. We can solve for $\overrightarrow{V}$ in the least squares sense as:

$$\overrightarrow{v} = (A_{3n}^T W_{3D}^2 A_{n3})^{-1} A_{3n}^T W_{3D}^2 A_{n3} B_{n1}, \tag{2.17}$$

where we have again introduced a diagonal matrix W (perhaps computed using 3D Gaussian values). We use a W with all its diagonal elements set to 1.0 in our work, as we see little accuracy improvement using W computed in other ways for Doppler data. We perform eigenvalue/eigenvector analysis of the integration matrix $A^T W^2 A$ to compute 3 eigenvalues $\lambda_3 \geq \lambda_2 \geq \lambda_1 \geq 0$ and accept as reliable full 3D velocities those velocities with $\lambda_1 > \tau_D$. We denote the least squares velocity as $\vec{V}_{ls} = (U_{ls}, V_{ls}, W_{ls})$ below.

Similarly, the 2D Horn and Schunck method can easily be extended into 3D. The 3D cost (energy) functional to be minimized becomes:

$$\begin{aligned} F = \int\int\int & (I_x U + I_y V + I_z W + I_t)^2 + \\ & \alpha^2 (U_x^2 + U_y^2 + U_z^2 + V_x^2 + V_y^2 + V_z^2 + W_x^2 + W_y^2 + W_z^2) \partial x \partial y \partial z. \end{aligned} \tag{2.18}$$

Again, the full velocity can be obtained iteratively by solving the Euler-Lagrange equations using the Gauss-Seidel method [7]. The iterative equations resulting for $(U, V, W)^T$ are:

$$U^{k+1} = \bar{U}^n - \frac{I_x \left[I_x \bar{U} + I_y \bar{V} + I_z \bar{W} + I_t \right]}{(\alpha^2 + I_x^2 + I_y^2 + I_z^2)}, \tag{2.19}$$

$$V^{k+1} = \bar{V}^k - \frac{I_y \left[I_x \bar{U} + I_y \bar{V} + I_z \bar{W} + I_t \right]}{(\alpha^2 + I_x^2 + I_y^2 + I_z^2)}, \tag{2.20}$$

$$W^{k+1} = \bar{W}^k - \frac{I_z \left[I_x \bar{U} + I_y \bar{V} + I_z \bar{W} + I_t \right]}{(\alpha^2 + I_x^2 + I_y^2 + I_z^2)}. \tag{2.21}$$

where α was typically 1.0, 10.0 or 100.0 and the maximum number of iterations was typically 50, 100 or 200 (the iterations could be stopped earlier if the L2 norm of the difference in two adjacent optical flow fields was less than some preset threshold τ (usually 0.001 or 0.00001).

2.1.2.3 3D Velocity Retrieval using Optical Flow

Based on the discussion above, (Xiaomei) Chen *et al.* [13, 14, 15] introduced the optical flow approach to solve the 3D full velocity retrieval problem for single Doppler radar datasets. The Doppler radar measures the radial component of full velocity on the radial beam and we replace normal velocity with radial velocity in our formulations. The full velocity $\overrightarrow{V} = (U, V, W)$ and the radial velocity V_r satisfy the relationship of 3D motion constraint equation (where $\vec{r}$ now replaces $\vec{n}$):

$$Ur_x + Vr_y + Wr_z = V_r. \tag{2.22}$$

$\hat{r} = (r_x, r_y, r_z)$ is the unit radial vector representing the radial direction. Using radial velocities we can see that the 3D Lucas and Kanade and 3D Horn and Schunck methods can be used unchanged.

Lucas and Kanade becomes:

$$\vec{V} = [A^T W^2 A]^{-1} A^T W B, \tag{2.23}$$

where, for $N = n \times n \times n$, we have:

$$\begin{aligned} A &= [\hat{r}_1, ..., \hat{r}_N], && (2.24)\\ W &= \text{diag}[W(x_1, y_1, z_1), ..., \ W(x_N, y_N, z_N)], && (2.25)\\ B &= (V_{r1}, ..., \ V_{rN}). && (2.26) \end{aligned}$$

We denote the least squares velocity as $\vec{V}_{ls} = (U_{ls}, V_{ls}, W_{ls})$ below.

Horn and Schunck becomes [6, 7, 70]:

$$\begin{gathered} \sum_R (\vec{V} \cdot \hat{r} - V_r)^2 + \alpha^2 \Big[(U_x)^2 + (U_y)^2 + (U_z)^2 + \\ (V_x)^2 + (V_y)^2 + (V_z)^2 + (W_x)^2 + (W_y)^2 + (W_z)^2 \Big], \end{gathered} \tag{2.27}$$

where U_x, U_y, U_z, V_x, V_y, V_z, W_x, W_y and W_z are the partial derivative of U, V and W with respect to x, y and z respectively. Iterative Gauss-Seidel equations that solve the Euler-Lagrange equations derived from this functional are used to compute (U, V, W). α

was typically set to 1.0 or 10.0 and the number of iterations was typically was limited to be 150 (with the iterations stopped earlier if convergence as measured by the norm of the difference in adjacent flow fields being less that $\tau = 10^{-3}$) is achieved.

(Xiaomei) Chen [13] found that using the regular Horn and Schunck global regularization has problems as the radial velocities already satisfies this smoothness constraint. Therefore they modified the smoothness constraint by introducing local least squares as the third constraint into this cost function. Now the cost function is:

$$\begin{aligned} F = \int\int\int & (\overrightarrow{V} \cdot \hat{r} - V_r)^2 + \\ & \alpha^2 (U_x^2 + U_y^2 + U_z^2 + V_x^2 + V_y^2 + V_z^2 + W_x^2 + W_y^2 + W_z^2) + \\ & \beta^2 ((U - U_{ls})^2 + (V - V_{ls})^2 + (W - W_{ls})^2) \partial x \partial y \partial z, \end{aligned} \tag{2.28}$$

where the U_{ls}, V_{ls}, W_{ls} are the pre-calculated least squares full velocities.

Figures 2.1 and 2.2 show a comparison of the full velocities retrieved using these two approaches on NEXRAD I data. The full velocity is represented by the white directed arrows. The length of the arrow reflects the magnitude, and the arrows shows the UV flow directions only. It can be seen that the Chen *et al.* regularization method provides a smoother flow field than the one based on the least squares method only.

2.2 Retrieving 3D Velocity using Dual-Doppler Radar

Due to limitations from using a single Doppler radar to recover 3D information, some later research has focused on using two overlapping Doppler radars (the Dual-Doppler radar). Armijo [1] first proposed a method to generate the full wind field of severe storms using Dual-Doppler radars.

2.2.1 The Coordinate System Conventions and Definitions

The coordinate system used in dual-Doppler radar techniques can mainly be divided into two categories: the right-handed rectangular Cartesian x, y, z coordinate system (Figure

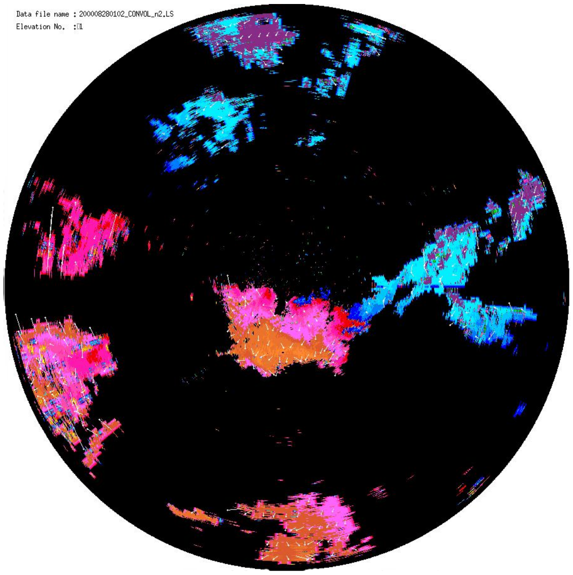


Figure 2.1: Retrieved full velocity by least squares method on radar data (2000, August 28th, 01:02)

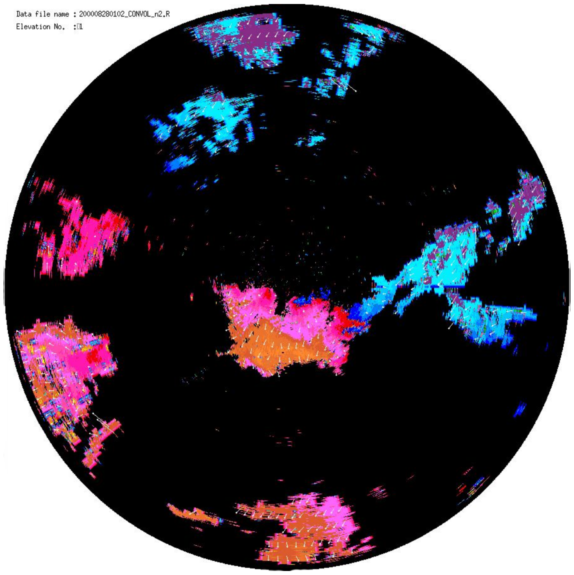


Figure 2.2: Retrieved full velocity by Regularization method on radar data (2000, August 28th, 01:02)

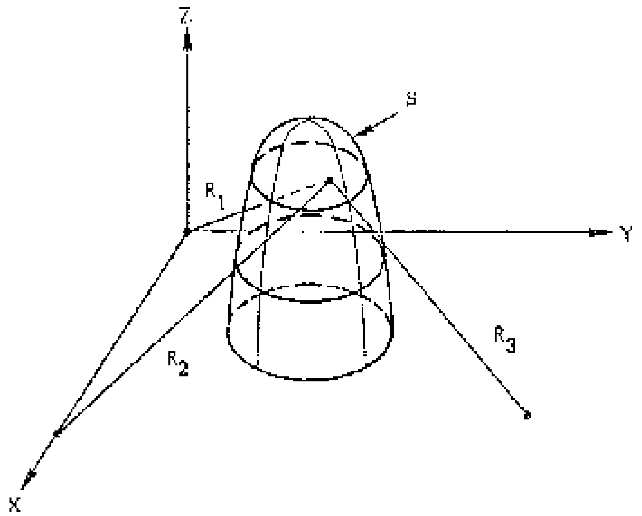


Figure 2.3: Cartesian Coordinate System, cited from Armijo's paper [1]

2.3) and the co-planar cylindrical polar coordinate system as shown in Fig. 2.4. Compared to the traditional Cartesian system, the co-planar coordinate system seems more appropriate due to the structure of Dual-Doppler radar systems.

Figure 2.3 shows us the basic Cartesian coordinate system, where two radar's locations are (X_1, Y_1, 0) and (X_2, Y_2, 0), then the correspond radial velocity detected at every point for each radar could be represented as:

$$V_1(x,y,z) = \frac{1}{R_1}[(x-X_1)u + (y-Y_1)v + z(w+V_t)] \tag{2.29}$$

$$V_2(x,y,z) = \frac{1}{R_2}[(x-X_2)u + (y-Y_2)v + z(w+V_t)] \tag{2.30}$$

where

$$R_1 = \sqrt{[(x-X_1)^2 + (y-Y_1)^2 + z^2]} \tag{2.31}$$

$$R_2 = \sqrt{[(x-X_2)^2 + (y-Y_2)^2 + z^2]}. \tag{2.32}$$

In these calculations, V_t is the negative terminal fallspeed of precipitation particles. $\overrightarrow{V} = (u, v, w)$ is the full velocity of wind field we need to retrieve. Since there are two

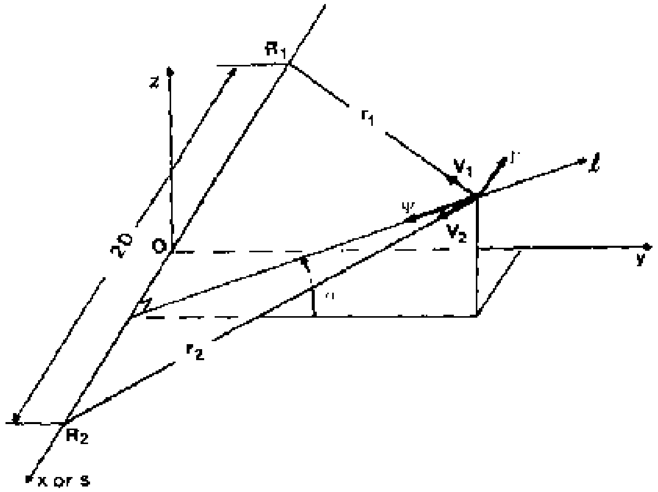


Figure 2.4: Coplanar Cylindrical Coordinate System, from Testud and Chong [71]

equations there are four unknown variables we need to introduce additional constraints to make the problem well-posed.

As an alternative to the Cartesian systems, Figure 2.4 depicts the basic structure of the Dual-Doppler case in co-planar cylindrical coordinate system. Assuming the locations of two Doppler radar are at (-D, 0, 0)and (D, 0, 0), in the so called α-plane, which is determined by the two radial vectors of radars. Any specific 3D environmental point can be located as (s, l, α) as shown in Figure 2.4, where s is a position along the x axis. The full velocity at the point (s, l, α) now can be expressed as $\overrightarrow{V} = (\Gamma, \Psi, \Phi)$ where Γ and Ψ can be calculated as:

$$\Gamma(s, l) = (r_1 V_1)/2D \tag{2.33}$$

$$\Psi(s, l) = [r_2(s + D)V_2 - r_1(s - D)V - 1]/(2lD) + V_t \sin\alpha. \tag{2.34}$$

Armijo [1]) has proposed some reliable approaches to estimate the magnitude of V_t. The third component of full wind velocity, Φ is the part of the velocity perpendicular to the α-plane (which is defined by the elevation angle α) and requires extra constraints to compute such as boundary conditions in Chong and Testud [18].

2.2.2 Early Doppler Work

Most modern Dual-Doppler radar investigation can be traced back to research by Armijo [1] in 1968, where he used simple formulas to calculate the horizontal velocity in cylindrical polar coordinates. In order to retrieve the vertical component of wind field, a widely-applied method in meteorology called the anelastic mass conservation equation (Equation 2.35) is used along with the projections of velocity along the direction of two Doppler radars:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial \omega}{\partial z} = \kappa\omega \tag{2.35}$$

In this case, the vertical velocity is obtained by integrating a quasi-horizontally mass flux divergence along the coplane azimuthal coordinate lines. The formula is given as:

$$\frac{\partial w}{\partial \alpha} = (kl\cos\alpha - \tan\alpha)w + \beta(s,l,\alpha)l\cos\alpha, \tag{2.36}$$

where

$$\beta(s,l,\alpha) = -\left(\frac{\partial \Gamma}{\partial x} + \frac{\partial \Psi}{\partial y}\right). \tag{2.37}$$

The Γ and Ψ variables are defined as above. Therefore, the solution for w, which satisfies the natural initial condition $w = 0$ on plane $\alpha = 0$, is given as:

$$\omega(s,l,\alpha) = \int_0^{\alpha} \beta(s,l,\theta)l\cos\alpha e^{kl(\sin\alpha - \sin\theta)}d\theta. \tag{2.38}$$

To solve this equation, Armijo assumes the terminal fallspeed V_t can be calculated using a known negative-valued function (not given), which however is not always true in reality. Additional practical experiments are also required to verify its reliability.

Ray and Doviak *et al.* [58] have implemented an approach to analyze a real tornado-like storm that occurred in April 1974. They computed the velocity components by combining terminal velocity estimates with the anelastic mass conservation equation. They estimate the terminal velocity from interpolating reflectivity:

$$\overline{V}_t = 2.6\overline{Z}^{0.107}(\frac{\gamma_0}{\gamma})^{0.4}. \tag{2.39}$$

where γ is the height-dependent air density. They also use a integration process to solve the anelastic mass conservation equation. This integration is based on the original method suggested in [1], which estimated the vertical velocity accumulatively from the ground air where the terminal velocity was assumed to be 0. They also implemented a Cressman filter (Cressman [21]) into the interpolation procedure to smooth the results. They claimed that by using all these techniques, the cyclonic circulation of tornado, and also the regions of up- and down-drafts are clearly evident.

Testud and Chong [18, 19, 71], worked on several factors which led to large errors in the Armijo solution. These factors include the non-simultaneous nature of data acquisition, spatial interpolation errors, discreteness errors and contamination of radial wind data. They also proposed a series of approaches to deal with these errors and to estimate the output variances. First, they filtered the raw data and interpolated data based on both data points and mathematical regularity by forcing continuous and differentiable up to 2^{nd} order. They also suggested that the minimization of the temporal errors can be done by calculating the advection velocity described by Gal-Chen [30]. Last they have discussed about the different boundary conditions used in the integration procedure to retrieve vertical velocity. The integration procedure can be processed in two directions: one is processing upwards from the ground level and the other one is processing downwards from the upper boundary in the highest available elevations. Then, an “optimized” solution was suggested with a “floating” ground level, where the ground condition $w_0 = 0$ is guaranteed. This “floating” ground level z_{opt} could be calculated as $z_{opt} = z_{max} - 0.35H$, where z_{max} is the maximum altitude and H is the air density scale height.

Doviak and Ray [24] described in detail how to estimate the errors in wind fields derived from Dual-Doppler radar measurements. They pointed out that the vertical velocity variance at high altitudes is related to an accumulation of errors due to the integration of the anelastic mass conservation equation. The biased errors in the integration process can keep increasing and lead to severe overall errors. To reduce this biased error, O'Brien [54] introduced a new algorithm which imposed vertical velocity values on both the upper and lower boundaries and then adjusted the horizontal divergence variationally in order to satisfy the anelastic mass conservation equation (Equation 2.35). Using this method, they differentiated the original mass conservation equation by height, and actually obtained the vertical velocity as the solution of mass conservation equation in a second-order format. Later this particular method has been extended to the variational framework Shapiro and Mewes [62].

2.2.3 Variational Method

Inspired by O'Brien [54], Shapiro and Mewes [62] proposed a native solution for the retrieval of full wind field from Dual-Doppler radars. In contrast to adopting the mass conservation equation directly, they applied it in its second order format and implement it into a Euler-Lagrange equation. Generally, their technique is equivalent to imposing the mass conservation equation as a constraint, which can be solved easily in a least squares framework. They found this technique is able to generate more flexible results, since it is also possible to implement more specialized constraints in their solution, such as radial velocity data, which leads to a generalized solution suitable for more complicated situations. They apply this technique directly using Cartesian coordinate system. The constraints they chose can be categorized as "Strong" or "Weak", depending on whether they are forced to be satisfied. According to whether the mass conservation and radial wind observation constraints are implemented as strong constraints or as weak constraints, the technique can be applied in three different cases: the mass conservation constraint is strong and the radial velocity constraint is weak, the mass conservation constraint is weak and radial velocity constraint is strong or both constraints are weak. They examined each algorithm in detail for these three formulations and provided the methods to calculate full velocities in each case.

Gao *et al.* [31] discussed the possibility of introducing more factors into the variance formulation of Shapiro and Mewes [62]. In addition to using the mass conservation equation and radial velocity observations, they adopted two other constraints for wind field: background and smoothness constraints.

In more detail, they try to minimize a cost function, J, which is the sum of squared errors due to the misfit between observation and their analysis subject to constraints. The formulation of J can be written as:

$$J = J_O + J_D + J_B + J_S, \tag{2.40}$$

where

$$\begin{aligned}
J_O &= \lambda_o(V - V_r) && (2.41)\\
J_B &= [\lambda_{ub}(u - u_b)^2 + \lambda_{vb}(v - v_b)^2 + \lambda_{wb}(w - w_b)^2] && (2.42)\\
J_D &= \lambda_d(\frac{\partial\overline{\rho}u}{\partial x} + \frac{\partial\overline{\rho}v}{\partial y} + \frac{\partial\overline{\rho}w}{\partial z})^2 && (2.43)\\
J_S &= [\lambda_{us}(\nabla^2 u)^2 + \lambda_{vs}(\nabla^2 v)^2 + \lambda_{ws}(\nabla^2 w)^2]. && (2.44)
\end{aligned}$$

J_O is the difference between the analyzed radial velocity and the observed radial velocity; J_D imposes a weak anelastic mass constraint on the wind field; J_B measures how close the analyzed data is to the background fields; and J_S represents the smoothness between analyzed data. The λ in the equations above are given different values representing weights assigned to each constraints. However through their experiment, it turns out that the results are not very sensitive to the precise values that they chose. They applied this method to both synthetic data and a real well-analyzed supercell storm. Due to the extra constraints they adopted here, they found that their method is less sensitive to the specification of boundary conditions and was able to show the correct circulation even when a few data were contaminated.

2.3 Windprofiler Radar

Other than the traditional Doppler radars that we've discussed above, there is another kind of pulsed radars, called Windprofilers, first developed in 1973, which are typically used to detect the motion of wind flow. Starting in 1983, the Wave Propagation Laboratory (WPL) [74] in Boulder, Colorado has continuously operated a network of windprofiler radars to measure hourly-averaged vertical profiles of the wind field. The laboratory implemented two kinds of windprofilers in the VHF and UHF frequency ranges. For economic and historic reasons, the frequency adopted for windprofiler ranges from as low as 40 to as high as 3000 MHz. The frequency chosen depends on the object being detected. A radar signal at a VHF frequency of 50 MHz reflects from large scale turbulence in clear air, heavy precipitation and hail but does not not reflect off the clouds well. The low frequency windprofiler also has some difficulties in retrieving data in the first kilometers (the lower atmosphere) [28]. The UHF frequency of 915 MHz detects clouds, light and heavy rain more easily than clear air turbulence. But it is more sensitive to hydrometeors[1], which could cause serious problems in estimating the vertical motion of wind. The 405 MHz radar can detect wind field in various light-to-heavy rain conditions. Hocking [34] provides a general summary of the applications of windprofiler radars all over the world. Specifically, Hocking [35, 36] presented a detailed introduction to the design and implementation of the O-Qnet (Ontario and Quebec, Canada) radar system, whose data we use in this thesis.

2.3.1 Data Processing

Strauch *et al.* [67] gives a brief description of the windprofiler structure as shown in Figure 2.5, which demonstrates the DBS (Doppler beam-swinging) method. The windprofiler has an array of antennas of four phased beams pointing to the opposite azimuths which are inclined about 15 degrees off the vertical direction. The first two are called the zonal (East-West direction) and the second two are meridional (North-South) beams, respectively.

---

[1] Hydrometeors are defined as any water or ice particles that have formed in the atmosphere or at ground level due to condensation. Water or ice particles blown from the ground into the atmosphere are also classed as hydrometeors. Some well-known hydrometeors include clouds, fog, rain, snow, hail, dew, rime, glaze, blowing snow, and blowing spray. This definition is taken verbatim from `www.britannica.com/EBchecked/topic/278940/hydrometeor`

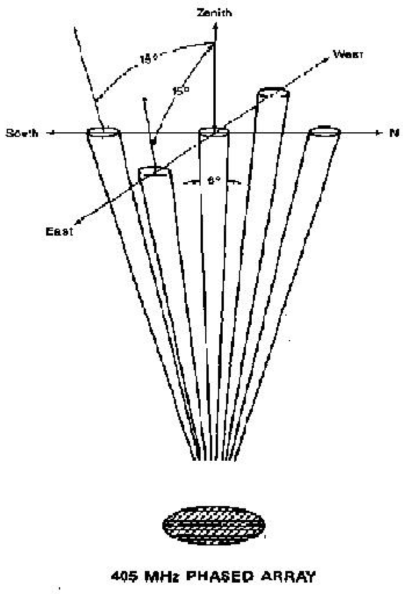


Figure 2.5: Antenna beam configuration for the 5-beam UHF wind profiler, from Strauch *et al.* [67].

There can be an extra zenith beam in the middle, which provides corrections of vertical motion. The distance between the four lateral beams and the vertical beam at a height of 10 km above the ground is about 2.7 km. The radar swings the beam in different directions in turn to determine the wind vectors. The resolution of the windprofiler is (originally) limited to 150 m by the existing hardware. May and Strauch [52] have discussed the signal processing algorithms used in the windprofiler, such as the First-Moment algorithm, and the random sample consensus algorithm.

The detection of the windprofiler antennas makes an assumption of data uniformity for the profiler data analysis. It is assumed the data detected on each beam can be considered as being retrieved simultaneously, i.e. the wind field is uniform over the scanned area of radar antenna beams. Therefore the five-beam profiler gives two independent estimates of horizontal wind components. The radial velocity measured on each antenna is given by

Equation (2.46):

$$\begin{aligned} V_{r_n} &= v\cos\theta + W\sin\theta + \delta V_{r_n}, \\ V_{r_e} &= u\cos\theta + W\sin\theta + \delta V_{r_e}, \\ V_{r_s} &= -v\cos\theta + W\sin\theta + \delta V_{r_s}, \\ V_{r_w} &= -u\cos\theta + W\sin\theta + \delta V_{r_w}, \\ V_{r_z} &= W + \delta V_{r_z} \end{aligned} \tag{2.45}$$

where the subscripts n, e, s, w, z represent the different antenna beam directions. W is the vertical velocity, assumed to be uniform across all antenna beams. Based on this equation, the horizontal and vertical wind velocities can easily be retrieved. Strauch *et al.* [67] compared the standard deviations of the estimated errors from the cases where the wind field is retrieved with and without a vertical correction. They found that the vertical correction of of the estimates from the zenith antenna beam significantly improved the relative accuracy of horizontal velocities, especially when the wind velocity magnitude was large. When vertical correction was involved, the measurement errors are the only limiting factors if these measurements are obtained with the highest signal-to-noise ratios. After choosing an appropriate signal-to-noise ratio, the precision of horizontal wind estimates can also be enhanced.

Alternatively in 1997, Hocking [34] presented the implementation of windprofiler in the CLOVAR (London, Ontario) VHF atmospheric radar network using the “space antenna” method, in which the antenna is fixed and doesn’t need to swing. The system is designed as a cross array covering roughly 4400 m^2. In the cross field, they used a fixed “quartet” equipped with 4 dipoles at the corners of a square, as shown in Figure 2.6. This method is believed to be more flexible and accurate.

Ecklund *et al.* [28] gave a basic description of the results that a windprofiler can provide. Their discussion is based on the weather atmosphere situation where the windprofiler is used, such as in *clear sky conditions*, *snow/rain conditions*, *thundershowers* or *tropical conditions*. They concluded that as the precipitation increases, the accuracy of the windpro-

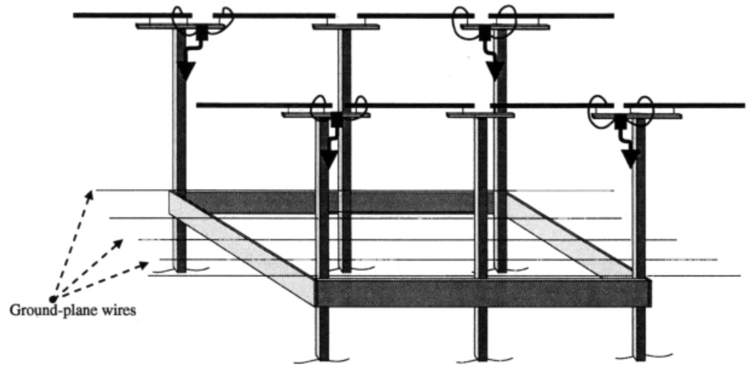


Figure 2.6: Schematic diagram of a typical "quartet", equipped with 4 dipoles, from Hocking [34].

filer measurements decreases. They also present the potential optimum solutions of using different antennas for different weather conditions, such as whether to use fixed-panel antennas or mechanical steered ones.

Wuertz *et al.* [74] investigated the effects of precipitation on the performance of the UHF windprofiler. They proposed examining the consensus of sub-hourly detection from different beams to estimate the precision of windprofiler data. Their discussion is divided into four categories due to the precipitation situations: *clear air*, *uniform precipitation*, *time-varying precipitation* and *space-varying precipitation*. Different measurement methodologies have been advised for each situation. They suggested that the correction of vertical motion should be done on a sub-hourly basis, before calculating the average consensus in the heavy precipitation case.

2.3.2 Applications of the Windprofiler Radar

Due to its unique observation features, there have been a wide variety of applications of windprofiler radars. Ecklund *et al.* [28] made the first attempt to measure the boundary layer of the atmosphere using a windprofiler. They developed a network of windprofilers to provide information in the boundary layer as well as in the troposphere and lower stratosphere. Other than the ordinary wind observations in fair weather conditions, windprofiler radars have also been used in extreme convective conditions. May and Rajopadhaya [51] observed a well-developed tropical squall line. The observed vertical motion and precipitation characteristics of this squall line are discussed in detail. These characteristics include the mean acceleration above the freezing level, the deep descent in the transition region and the up- and down-draft couplets in the stratiform region (all of which fit well with the widely-accepted conceptual squall line model).

Other research investigated the use of windprofilers for a variety of purposes. Gage *et al.* [29] adopted a 915-MHz UHF windprofiler to analyze tropical convective cloud systems. Long term observations were used to develop the climatology of precipitating cloud systems at windprofiler radar sites in the tropics. Lucas *et al.* [49] proposed an algorithm to identify the frontal zones in windprofiler data. This identification was based on the temperature gradient, using a quasi-geostrophic thermal wind retrieval technique. Sengupta and Watson [60] worked on the detection of tropospheric ducts using 915-MHz windprofiler, where a duct is the radar beam occurring when the humidity and temperature change rapidly with height.

2.3.3 The Combination of Windprofiler and Doppler Radar

In order to improve the monitoring capabilities of the windprofiler radar, an appealing application is the synergetic cooperation with several other ground-based instruments. Bianco *et al.* [8] implemented a combination of a windprofiler radar and a microwave radiometer. They proposed a self-consistent remote sensing method to provide high resolution atmosphere humidity measurements.

Yoe *et al.* [81] first compared the data quality of horizontal and vertical motions from a VHF windprofiler with that from a Doppler radar. In particular, they analyzed the data processing algorithm of a VHF windprofiler and noticed a double peaked spectra in the observations caused by precipitation or lightning. In order to apply VHF in complicated convective conditions, they advised two methods (First-Moment and Least-Squares methods) to eliminate different kinds of outliers and estimate an average velocity. Two Doppler radars located nearby provided a comparison of the same flow using the EVAD method described above. There is generally good agreement between the magnitudes of vertical velocities measured by these instruments from 5 km to 10 km. However, the VHF radar estimated a much higher magnitude of downdraft near the melting layer at round 3 km. Yoe *et al.* acknowledge that this comparison is limited in accuracy and is incomplete, and the understanding of these large discrepancies requires further research.

Cifelli *et al.* [20] incorporated a VHF windprofiler into their MCSs (mesoscale convective systems) experimental system, where several single-Doppler radars alone were being used. Their analysis includes two extensions of the VAD and EVAD methods and the introduction of the CEVAD (concurrent EVAD) method and the traditional VVP method. The strengths and weaknesses of Doppler radar and windprofiler radar were presented. Compared to a Doppler radar, the windprofiler radar is able to provide measurements in a relatively small domain directly, regardless of the weather conditions. Although these measurements may be complicated and noisy, boundary conditions were not required in their analysis.

The windprofiler radar measurements showed larger gradients and magnitudes in horizontal divergence as well as vertical motions. Their measurements could be made at significant distance over the top of the single Doppler radar. Although it is hard to quantify the differences between Doppler radar and windprofiler radar measurements, it is obvious that the horizontal divergence distributions of the single Doppler radar case are much narrower. The vertical motions measured by windprofiler and these single Doppler radar techniques showed qualitatively good agreement, but a detailed comparison at any given height is very limited (the quality of Doppler and Windprofiler radar vertical velocity measurements is subject to an on-going debate).

Furthermore, the relatively small scale detection provided by a windprofiler radar may reveal the small vertical circulations occurring in the stratiform region, which have a significant impact on precipitation growth. They concluded that windprofiler radars could work well as a complementary tool with a single Doppler radar.

More recently, Lothon *et al.* [46] presented a clear air atmospheric boundary layer (ABL) wind profiling aspect of the C-band Doppler radar (radars that operate on a wavelength of 4-8 cm and a frequency of 4-8 GHz) and compared it with the data retrieved by two UHF windprofilers radars. They adopted the VVP method [72] for the analysis of a Doppler radar and the two windprofiler radars. The windprofiler radars has a 75 m vertical resolution and 5 min temporal resolution. They concluded that the Doppler radar with its large scanning coverage had significant advantages over VHF windprofilers for observing the atmosphere in 3D. The Doppler radar covers a larger observation volume and retrieves more data at lower elevation levels. The difference between the maximum reflectivity values that are obtained by these radars is determined by the different backscattering sources that deflect the radar signal back to the radar. Comparing the estimated vertical motions from the two radars, they found a persistent mean downward velocity of about $-0.3\ ms^{-1}$ in the average vertical profile for a long term experiment, which contradicts the general accepted vertical velocity distribution. On the other hand, another view of vertical transport offered by the C-band Doppler radar shows the wind has a mainly upward velocity. Therefore, they argue that this evidence indicated that the accuracy of windprofilers in this dimension cannot be trusted. There have been some arguments against their conclusion. For example, Worthington [73] has presented a number of explanations for these results and commented that their conclusions may be too narrowly focused since the specific geography location near the original experiment could cause the failure instead of the experiment itself. Later, Lothon *et al.* [47] explained their experiment in more detail and renewed their conclusions. Nevertheless, further investigation using more accurate devices and in various less-favorable conditions is necessary in order to resolve this debate.

2.4 Detection and Tracking of Severe Storms

A major application of Doppler radar is to predict the incoming severe weather storms based on the reflectivity information. There are two key processes to solve this problem: first recognize the storm, measure its size and location/center and second track its trajectory over time. When it is necessary, a warning of potential damage should be issued. Yilmaz, Javed and Shah [80] presented a general survey of all the mainstream algorithms applied in object tracking in Computer Vision. They discuss the various steps of the tracking procedure, such as object representation, feature selection and detection, and object tracking. The various tracking algorithms can be classified according to the object models they are using as point tracking, kernel tracking and silhouette tracking. Compared to the other popular tracking objects, for example the human face, storms are only clusters of connected data points with no shape character for feature extraction. Therefore the traditional point tracking algorithm such as the "Hungarian method" [42] is widely applied in this area by NOAA and Environment Canada.

Barron and Mercer, with their students have worked in this area since the early 1990's. The 2D work was summarized by Barron *et al.* [4]. In the early 2000's, they proposed a method to detect and track the 3D storms using interval arithmetic (called "fuzzy" algebra) to represent deformable objects (severe storms) [53] and track them in an incremental relaxation labeling algorithm [6].

2.4.1 Storm Detection using a Flood Fill Algorithm

The severe weather storm in Doppler radar imagery is represented by high density precipitation [59]. Therefore, when detected by a Doppler radar, a storm can be considered as a cluster of data points that have a high reflectivity value. The first step in detecting a storm is to cluster voxels with high reflectivity values into one (or a few) clusters. If the size of a cluster containing storm points is greater than some threshold, then this cluster is considered a storm. An algorithm to detect adjacent storm points is based on a recursive **Flood**

Fill algorithm [4, 5]. This algorithm marks a voxel as a part of a storm if it has a high reflectivity value and then applies this same process recursively to all its neighboring voxels. In this way, 3D storms can be detected by collecting all connected neighboring voxels together into a 3D volume. The storm reflectivity threshold was set to 30 - 35 dBZ, as Qiu [56] and Tang *et al.* [68] did, which is widely accepted in other Meteorology research as Johnson *et al.* [38] suggested.

2.4.2 Storm Representation by Fuzzy Point Algebra

Cheng *et al.* [17] first used this flood fill algorithm in 2D. They adopted fuzzy point algebra (interval arithmetic) [53] and represented a storm as a 2D fuzzy point. A fuzzy point is actually a circle specified by a radius and origin coordinates. Any set of coordinates inside this circle is said to match the fuzzy point. Storms were represented by fuzzy points and fuzzy vectors were used to represent fuzzy point motion between adjacent Doppler datasets. Using this notion of fuzzy vectors, the compatibility of adjacent fuzzy vectors (using measures as fuzzy magnitude and fuzzy angle definitions with interval arithmetic ranges) was determined and used in a 2D relaxation labeling tracking algorithm to compute storm trajectories. Because this work was 2D, it was restricted to single low elevation Doppler radar datasets. A circle does not always capture a Doppler storm (especially oblong storms) and sometimes, as a result, the storms oscillate with a potential track. Qiu's M.Sc. thesis [56] extended fuzzy points into 3D. 3D fuzzy point algebra is now defined and applied to all elevations in 3D Doppler datasets and a 3D tracking algorithm was devised using this tool. Tang [68, 69, 70] noted that the fuzzy point extension of a 2D circle into a 3D sphere was not as appropriate as using a 3D ellipsoid. She extended 3D fuzzy algebra to use these 3D fuzzy storms by adding extra mechanisms to model the orientation of ellipsoids (not needed for 3D spheres) in the tracking algorithm. This new solution demonstrated superior tracking results. A similar representation model is also adopted by Storlie, Lee, Hanning and Nychka [22], Dixon and Wiener [66].

2.4.3 Tracking Storm by Relaxation Labeling Algorithm

Although the detection and representation of storms are simpler compared to the other tracking topics, the tracking of storms could be very challenging in some complicated cases where storm shapes are highly deformable. Different from the traditional tracking problems where objects could block each other and cause feature extraction failure [33], storms are considered as moving in the same horizontal plane that they could merge into one super storm and also split into several smaller parts. During the 1990's, Dixon and Wiener [22], is to the author's knowledge, the first attempt to enhance the original tracking algorithm by handling merging and splitting of storms correctly. They based their solution on the combinational optimization of all the possible paths of storms. More recently, Storlie, Lee, Hanning and Nychka [66] have furthered the discussion of storm tracking in a frequent merging and splitting situation by using a statistical method called modified MHT (Multiple Hypothesis Tracking) method. Their research primarily worked on the recognition of merging and splitting only, other than generating smooth tracks based on this information. Besides, their algorithm focused on the enhancement of tracking results in a large coverage area such as the whole North America territory, compared to the smaller and more detailed experiment environment we use around the Great Lakes area. More recently Baldwin [2] proposed a verification framework to measure the precision of storm forecasting. Xu and Chandrasekar [75] developed a storm motion estimation algorithm working in Fourier domain to give steady prediction of storms moving in a temporal sequence.

The storm tracking algorithm used in our work was originally developed for the 2D case [82] in 1991. Krezeski [41] proposed a temporal relaxation labeling algorithm to track multiple storm centers using Euclidean points to represent storms. Later Cheng [16, 17] improved this algorithm by using fuzzy storm centers in the 2D Doppler datasets. Qiu [56, 57] and Tang [68, 69, 70] then extended this method into 3D and this thesis builds on that work. We note that our tracking algorithm can use 3D optical flow as one of many compatibility functions [70] when determining if two storms belong to the same track or not. This work will be fully described in the following chapters. Our optical flow algorithms for single/dual/multiple Doppler radars and windprofiler radars is given in Chapters 3, 4 and 5 while our tracking algorithm is given in Chapters 6 and 7.

Part I

Velocity Retrieval with Multiple Radars

Chapter 3

Velocity Retrieval with Optical Flow

3.1 Introduction

In this chapter we present some methods to retrieve 3D full velocity from radial velocity data using Optical Flow techniques ([13, 14, 15, 69]). Optical Flow is a technology widely applied in the Computer Vision area. The 2D Optical Flow technique estimates the 2D motion of pixels in images taken from an image sequence. A simple 3D extension to 2D Optical Flow algorithms allows the computation of the 3D volumetric motion of voxels in a sequence of 3D volumes. In meteorology applications, 3D Optical Flow is a measure of the 3D wind velocity. Generally the solution of optical flow can be divided into two classes: one uses the least squares method [48] and the other uses the global regularization method [37]. Reliable performance based on these two solutions has already been obtained on real 3D Doppler radar data (NEXRAD *I*) in [14, 15]. Here we generalize their method and apply it to the multiple radar case.

3.2 General Multiple Least Squares Algorithm

As Barron *et al.* [6, 13] proposed, the full velocity of wind can be calculated with the optical flow method using the least squares approach using radial velocity data from one radar. The calculation is similar to solving the optical flow constraint equation Equation 2.12. To generalize this discussion to apply to a wider coverage area, here we extend their method to the overlapping multi-Doppler radar case so that the algorithm works properly in the multi-Doppler radars' overlapping area. It should be noted that the discussion of the formula focusses on the multiple-radar format. However, in the later experiment, we only present results using synthetic and real data for the Dual-Doppler radar case, that is, where only two Doppler radar overlap. This is due to the fact currently we don't have data available from an area that is covered by more than two Doppler radars.

Assume there are M Doppler radars, $R_1, R_2, \ldots, R_i, \ldots, R_M$, whose coverage areas overlap. In each radar's data, for each voxel (x, y, z), we record the radial velocity V_{r_i} from each radar, which is the radial component of full velocity on the beam direction of radar i. We denote the full velocity at a data point as $\vec{V} = (U, V, W)$ and radial velocity from radar R_i as V_{r_i}. Then the radial unit direction vector from the radar center of R_i is denoted $\hat{r}_i = (r_{x_i}, r_{y_i}, r_{z_i})$. The projection of the full velocity in the radial direction can be written as:

$$\vec{V} \cdot \hat{r}_i = U r_{X_i} + V r_{Y_i} + W r_{Z_i} = V_{r_i}, \tag{3.1}$$

The relationship between radial velocity and full velocity can be considered as an example of the optical flow constraint equation. To apply the retrieval process, we first smooth the radial velocity obtained from all the radars. Then at any given 3D position (x, y, z), we detect the number of available radars as m $(0 \leq m \leq M)$. In the dataset of every available radar we select a small neighbourhood, for example $3 \times 3 \times 3$ or $5 \times 5 \times 5$ around the vicinity of this point. Like Lucas and Kanade [48], since this local vicinity is very small compared to the whole Doppler dataset, we assume a local smoothness in this neighbourhood area, so all the points in the neighbourhood have the same full velocity $\vec{V}$. In order to retrieve full velocity from all data points in the multiple Doppler radar case, we parse each radar using its polar coordinate system. At each point, we gather together the data in the neighbouring area in the same radar and from other radars that have data in the overlapping areas. All these data are used in the velocity calculation. If there are N_i voxels in the selected neighbourhood area from radar R_i, we can set up a linear system of equations with an $N_i \times 3$ matrix on the left hand side and an $N_i \times 1$ vector on the right hand side described as:

$$\begin{bmatrix} r_{x_1} & r_{y_1} & r_{z_1} \\ r_{x_2} & r_{y_2} & r_{z_2} \\ \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots \\ r_{x_{N_i}} & r_{y_{N_i}} & r_{z_{N_i}} \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \approx \begin{bmatrix} V_{r_1} \\ V_{r_2} \\ \ldots \\ \ldots \\ V_{r_{N_i}} \end{bmatrix}, \tag{3.2}$$

where we use the ≈ symbol rather than the direct = symbol, because the radial velocities may have measurement error and certainly have representation errors (as they are “binned” to fit one byte storage locations). The solution of full velocity (U, V, W) will be a least squared approximation to all the equations.

The smoothness also applies to the overlapping radar datasets in the neighbouring region. So all the data in the vicinity should share the same full velocity but have different radial velocities $\vec{V}_{r_i}$. Now at a given data point (x, y, z), we have the total number of voxels from all the radars as $N = N_1 + N_2 + \ldots + N_m$ (N_i could be 0). Collecting all the data from various radars, we can build the $N \times N \times N$ matrix and generate an $N \times 3$ linear system of equations as:

$$\begin{bmatrix} r^1_{x_1} & r^1_{y_1} & r^1_{z_1} \\ r^1_{X_2} & r^1_{Y_2} & r^1_{z_2} \\ \dots & \dots & \dots \\ r^1_{x_{N_1}} & r^1_{y_{N_1}} & r^1_{z_{N_1}} \\ \dots & \dots & \dots \\ r^i_{x_1} & r^i_{y_1} & r^i_{z_1} \\ \dots & \dots & \dots \\ r^i_{x_{N_i}} & r^i_{y_{N_i}} & r^i_{z_{N_i}} \\ \dots & \dots & \dots \\ r^m_{x_1} & r^m_{y_1} & r^m_{z_1} \\ \dots & \dots & \dots \\ r^m_{x_{N_m}} & r^m_{y_{N_m}} & r^m_{z_{N_m}} \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \approx \begin{bmatrix} V^1_{r_1} \\ V^1_{r_2} \\ \dots \\ V^1_{r_{N_1}} \\ \dots \\ V^i_{r_1} \\ \dots \\ V^i_{r_{N_i}} \\ \dots \\ V^m_{r_1} \\ \dots \\ V^m_{r_{N_m}} \end{bmatrix}, \tag{3.3}$$

where $r^i_{x_j}$, $r^i_{y_j}$, $r^i_{z_j}$ are the three components of the radial direction unit from the j^{th} voxel of the radar R_i ($1 \leq i \leq m$ and $0 \leq j \leq N_i$), while $V^i_{r_j}$ is the j^{th} radial velocity from that radar. Now, the N data points are from different radars. How many radars are involved in the calculation depends on the exact position. This group of equations can be rewritten as:

$$A_{lsN\times3}\vec{V}_{3\times1} \approx B_{N\times1}, \tag{3.4}$$

which can be solved using the least squares method as:

$$A^T_{lsN\times3}A_{lsN\times3}\vec{V}_{3\times1} = A^T_{lsN\times3}B_{N\times1}, \tag{3.5}$$

where $A^T_{ls}A_{ls}$ is a symmetric real matrix (all eigenvalues are real and positive). This system can be solved if and only if $A^T_{ls}A_{ls}$ can be reliably invertible (which means the 3×3 matrix is non-singular). It admits the solution as:

$$\vec{V} = [A^T_{ls}A_{ls}]^{-1}A^T_{ls}B. \tag{3.6}$$

We can calculate the eigenvalues ($\lambda_0 \leq \lambda_1 \leq \lambda_2$) and their corresponding eigenvectors ($\hat{e}_0$, $\hat{e}_1$ and $\hat{e}_2$) from this 3×3 symmetric least square integration matrix $A_{ls}^T A_{ls}$. These eigenvalues are used to justify whether the matrix can be inverted reliably. Often it is suggested that these equations (in Equation (3.3)) be weighted as a function of their distance from (x, y, z) (perhaps by a Gaussian) but such weighting schemes have proved to have insignificant effect on the computed velocities and we have discontinued this practice.

3.3 General Multiple Regularization Algorithm

In the multiple Doppler radar case, the Least Squares solution provides full velocity results based on a local smoothness assumption. On the other hand, the regularization method generates full velocity based on the global minimization of various derivatives. We have devised a 3D regularization method based on an extension of Horn and Schunck's 2D optical flow regularization algorithm [37]. That is, a number of constraints on 3D velocity are minimized (regularized) over the 3D domain.

The first term we use is the 3D **Radial Velocity Constraint**, which requires that the full velocity projected in the radial direction be the radial velocity:

$$\vec{V} \cdot \hat{r} = V_r, \tag{3.7}$$

where $\vec{V} = (U, V, W)$ is the local 3D velocity (which we want to compute), $\hat{r}$ is the local unit radial velocity direction (which we know precisely from the structure of the radar data) and V_r is the measured local radial velocity magnitude.

The second constraint is a 3D Horn and Schunck-like **Velocity Smoothness Constraint**, which requires that velocity vary smoothly everywhere by keeping the velocity component derivatives in the 3 dimensions as small as possible.

Thirdly, the **Least Squares Velocity Consistency Constraint** is based on an extension of the 2D Lucas and Kanade least squares optical flow algorithm [48] into 3D, using the least squares method we described in the previous section. This constraint requires computed velocities to be consistent with local least squares velocities. As discussed in Chen

[13], the regularization method fails to provide good results without this least squares constraint since the raw radial velocity data already satisfy the first two constraints naturally. (Radial velocity itself is already smoothed, and if $\vec{V} = V_r \cdot \hat{r}$ then $\vec{V} \cdot \hat{r} - V_r = 0$.) Therefore it can only generate a velocity with the same magnitude as radial velocity in the radial direction. The additional retrieved flow using least squares method is integrated here to force the regularization method to produce full velocity in three directions. This follows from the work by Bruhn et al. [10], which first proposed the integration of the Lucas and Kanade and Horn and Schunck algorithms in 2D.

The regularization functional is the sum of these constraints:

$$\begin{aligned}&\int\int\int \underbrace{\sum_{i=1}^{m} \delta_i^2 (\vec{V} \cdot \hat{r} - V_{r_i})^2}_{\text{Radial Velocity Constraint}} + \\ &\alpha^2 \underbrace{(U_X^2 + U_Y^2 + U_Z^2 + V_X^2 + V_Y^2 + V_Z^2 + W_X^2 + W_Y^2 + W_Z^2)}_{\text{Velocity Smoothness Constraint}} + \\ &\sum_{i=1}^{m} \beta_i^2 \underbrace{((U - U_{ls_i})^2 + (V - V_{ls_i})^2 + (W - W_{ls_i})^2)}_{\text{Least Squares Velocity Consistency Constraint}}\end{aligned} \tag{3.8}$$

The Lagrange multipliers α, β, and δ_i (for each radar i) represent the relative weights of these various terms. The δ_is are Gaussian functions on the distance between the different radars and the current computational point. m $(0 \leq m \leq M)$ is the total number of available radars that are implemented.

Euler-Lagrange equations to minimize this functional are;

$$F_U - \frac{d}{dX} F_{U_X} - \frac{d}{dY} F_{U_Y} - \frac{d}{dZ} F_{U_Z} = 0, \tag{3.9}$$

$$F_V - \frac{d}{dX} F_{V_X} - \frac{d}{dY} F_{V_Y} - \frac{d}{dZ} F_{V_Z} = 0, \tag{3.10}$$

$$F_W - \frac{d}{dX} F_{W_X} - \frac{d}{dY} F_{W_Y} - \frac{d}{dZ} F_{W_Z} = 0, \tag{3.11}$$

where

$$F_U = 2\sum_{i=1}^{m} \delta_i^2(\vec{V}\cdot\hat{r} - V_r)r_1 + 2\sum_{i=1}^{m}\beta_i^2(U - U_{ls})), \quad (3.12)$$

$$F_V = 2\sum_{i=1}^{m} \delta_i^2(\vec{V}\cdot\hat{r} - V_r)r_2 + 2\sum_{i=1}^{m}\beta_i^2(V - V_{ls})), \quad (3.13)$$

$$F_W = 2\sum_{i=1}^{m} \delta_i^2(\vec{V}\cdot\hat{r} - V_r)r_3 + 2\sum_{i=1}^{m}\beta_i^2(W - W_{ls})), \quad (3.14)$$

$$F_{U_X} = 2\alpha^2 U_X, \quad (3.15)$$

$$F_{U_Y} = 2\alpha^2 U_Y, \quad (3.16)$$

$$F_{U_Z} = 2\alpha^2 U_Z, \quad (3.17)$$

$$F_{V_X} = 2\alpha^2 V_X, \quad (3.18)$$

$$F_{V_Y} = 2\alpha^2 V_Y, \quad (3.19)$$

$$F_{V_Z} = 2\alpha^2 V_Z, \quad (3.20)$$

$$F_{W_X} = 2\alpha^2 W_X, \quad (3.21)$$

$$F_{W_Y} = 2\alpha^2 W_Y, \quad (3.22)$$

$$F_{W_Z} = 2\alpha^2 W_Z, \quad (3.23)$$

$$
\begin{aligned}
\frac{dF_{U_X}}{dX} &= 2\alpha^2 U_{XX}, && (3.24)\\
\frac{dF_{U_Y}}{dY} &= 2\alpha^2 U_{YY}, && (3.25)\\
\frac{dF_{U_Z}}{dZ} &= 2\alpha^2 U_{ZZ}, && (3.26)\\
\frac{dF_{V_X}}{dX} &= 2\alpha^2 V_{XX}, && (3.27)\\
\frac{dF_{V_Y}}{dY} &= 2\alpha^2 V_{YY}, && (3.28)\\
\frac{dF_{V_Z}}{dZ} &= 2\alpha^2 V_{ZZ}, && (3.29)\\
\frac{dF_{W_X}}{dX} &= 2\alpha^2 W_{XX}, && (3.30)\\
\frac{dF_{W_Y}}{dY} &= 2\alpha^2 W_{YY}, && (3.31)\\
\frac{dF_{W_Z}}{dZ} &= 2\alpha^2 W_{ZZ}. && (3.32)
\end{aligned}
$$

Since $\nabla^2 U = U_{XX} + U_{YY} + U_{ZZ}$, $\nabla^2 V = V_{XX} + V_{YY} + V_{ZZ}$ and $\nabla^2 W = W_{XX} + W_{YY} + W_{ZZ}$ and expand $\vec{V} \cdot \hat{r}$ as $Ur_1 + Vr_2 + Wr_3$, we can rewrite the Euler-Lagrange equations as:

$$
\begin{aligned}
\sum_{i=1}^{m} \delta_i^2 (Ur_1 + Vr_2 + Wr_3) r_1 + \sum_{i=1}^{m} \beta^2 U &= \alpha^2 \nabla^2 U + \sum_{i=1}^{m} \beta^2 U_{ls} + \sum_{i=1}^{m} \delta_i^2 V_r r_1, && (3.33)\\
\sum_{i=1}^{m} \delta_i^2 (Ur_1 + Vr_2 + Wr_3) r_2 + \sum_{i=1}^{m} \beta^2 V &= \alpha^2 \nabla^2 V + \sum_{i=1}^{m} \beta^2 V_{ls} + \sum_{i=1}^{m} \delta_i^2 V_r r_2, && (3.34)\\
\sum_{i=1}^{m} \delta_i^2 (Ur_1 + Vr_2 + Wr_3) r_3 + \sum_{i=1}^{m} \beta^2 W &= \alpha^2 \nabla^2 W + \sum_{i=1}^{m} \beta^2 V_{ls} + \sum_{i=1}^{m} \delta_i^2 V_r r_3. && (3.35)
\end{aligned}
$$

As Horn and Schunck [37] suggested, $\nabla^2 U \approx \bar{U} - U$, $\nabla^2 V \approx \bar{V} - V$ and $\nabla^2 W \approx \bar{W} - W$. Note that $\nabla^2 U$, $\nabla^2 V$ and $\nabla^2 W$ are assumes to have the same units as U, V and W (and $\bar{U}$, $\bar{V}$ and $\bar{W}$) or to have the conversion factor between the units is 1. Horn and Schunck do not explicitly state this. Then the Euler-Lagrange equations are rewritten as:

$$
\begin{aligned}
&(\sum_{i=1}^{m}\delta_i^2 r_1^2+\alpha^2+\sum_{i=1}^{m}\beta^2)U+(\sum_{i=1}^{m}\delta_i^2 r_1 r_2)V+(\sum_{i=1}^{m}\delta_i^2 r_1 r_3)W=(\alpha^2\bar{U}+\sum_{i=1}^{m}\beta^2 U_{ls}+\sum_{i=1}^{m}\delta_i^2 V_r r_1),\\
&(\sum_{i=1}^{m}\delta_i^2 r_1 r_2)U+(\sum_{i=1}^{m}\delta_i^2 r_2^2+\alpha^2+\sum_{i=1}^{m}\beta^2)V+(\sum_{i=1}^{m}\delta_i^2 r_2 r_3)W=(\alpha^2\bar{V}+\sum_{i=1}^{m}\beta^2 V_{ls}+\sum_{i=1}^{m}\delta_i^2 V_r r_2),\\
&(\sum_{i=1}^{m}\delta_i^2 r_1 r_3)U+(\sum_{i=1}^{m}\delta_i^2 r_2 r_3)V+(\sum_{i=1}^{m}\delta_i^2 r_3^2+\alpha^2+\sum_{i=1}^{m}\beta^2)W=(\alpha^2\bar{W}+\sum_{i=1}^{m}\beta^2 W_{ls}+\sum_{i=1}^{m}\delta_i^2 V_r r_3),
\end{aligned}
\tag{3.36}
$$

or in matrix form as:

$$
\underbrace{\begin{bmatrix}
(\sum\delta_i^2 r_1^2+\alpha^2+\sum\beta^2) & (\sum\delta_i^2 r_1 r_2) & (\sum\delta_i^2 r_1 r_3)\\
(\sum\delta_i^2 r_1 r_2) & (\sum\delta_i^2 r_2^2+\alpha^2+\sum\beta^2) & (\sum\delta_i^2 r_2 r_3)\\
(\sum\delta_i^2 r_1 r_3) & (\sum\delta_i^2 r_2 r_3) & (\sum\delta_i^2 r_3^2+\alpha^2+\sum\beta^2)
\end{bmatrix}}_{A}
\begin{bmatrix} U\\ V\\ W\end{bmatrix}=
$$

$$
\begin{bmatrix}
(\alpha^2\bar{U}+\sum\beta^2 U_{ls}+\sum\delta_i^2 V_r r_1)\\
(\alpha^2\bar{V}+\sum\beta^2 V_{ls}+\sum\delta_i^2 V_r r_2)\\
(\alpha^2\bar{W}+\sum\beta^2 W_{ls}+\sum\delta_i^2 V_r r_3)
\end{bmatrix}.
\tag{3.37}
$$

We assign the initial guess of full velocities as $(0, 0, 0)$. Thus the equations can be solved using Gauss Seidel iterative method as:

$$
\begin{bmatrix} U^{n+1}\\ V^{n+1}\\ W^{n+1}\end{bmatrix}=A^{-1}
\begin{bmatrix}
(\alpha^2\bar{U}^n+\sum\beta^2 U_{ls}+\sum\delta_i^2 V_r r_1)\\
(\alpha^2\bar{V}^n+\sum\beta^2 V_{ls}+\sum\delta_i^2 V_r r_2)\\
(\alpha^2\bar{W}^n+\sum\beta^2 W_{ls}+\sum\delta_i^2 V_r r_3)
\end{bmatrix}.
\tag{3.38}
$$

It should be noted that there is risk that the equations do not stably converge. We cannot find find any recent discussion about this potential problem in the literature. In our experience, we have always obtained convergence. We always monitored the change in the velocity values between two adjacent iterations (the L_2 norm of the difference vector) to ensure that convergence is always occurring. We also used a preset maximum iteration number to terminate the iteration procedure eventually is our convergence criterion is not satisfied (never invoked).

3.4 Discussions

In this chapter, we proposed two methods in the multiple Doppler format to integrate the data from two or more radars to recover the full velocity from radial velocity data using Optical Flow techniques. The first one uses the least squares approach [48] and the other uses the global regularization method [37]. The least squares method is based on the local smoothness assumption, while the second regularization method works on a global energy function to minimize all the derivatives. Besides, the regularization method uses not only the radial velocity data but also implements the pre-calculated results from the least squares method to generate reasonable smooth results. In the next chapter we will design a synthetic experiment to simulate the 3D motion of wind, and use the generated artificial data to evaluate the performance of the dual radar retrieval methods compared with the single ones. Real data results will also be discussed.

Chapter 4

Synthetic and Real Data Experiment

4.1 Synthetic Experiment Design

In order to evaluate the claimed improvements to the calculation of full velocity provided by the Dual Doppler methods, a quantitative evaluation of the accuracy and robustness of the retrieved full velocity flow is necessary. Here we design a synthetic experiment where the ground truth of the 3D full velocities is already known. By comparing the estimated velocities from single and dual methods with the correct velocities, we can measure the performances of our recovery methods quantitatively and claim which one works the best.

Since the wind motion in meteorology can be very complex, we adopt different groups of synthetic data and furthermore add several levels of variation and noise to the synthetic data. It allows the performance of our methods to be evaluated exhaustively under various experimental scenarios. By variation, we mean changes to the preset magnitude and direction of the synthetic data. By noise, we mean the errors introduced by the radar detection procedure.

As in Chen's paper [13], this artificial experiment can be divided into the following steps:

1. Set the synthetic full velocity at each position in the dataset of each Doppler radar. The synthetic data is determined by two factors: the constant **base velocity** and the

changeable **variation velocity**.

2. Decide the radial velocity for each voxel based on its synthetic full velocity and the relative radial direction. This radial velocity calculated here is then “polluted” by noise in order to simulate the inevitable errors during radar detection.

3. Apply the least squares method or regularization method to the synthetic data and obtain the estimated full velocity flow. Both the single and dual radar methods will be tested.

4. Compare the estimated results with the correct velocity. Several quantitative analyses for the error estimation will be performed to evaluate their performances.

4.1.1 The Generation of Synthetic Velocity

Our synthetic data is determined by two factors: the constant **base velocity** and the changeable **variation velocity**:

$$\vec{V} = \vec{V}_{base} + \vec{V}_{var}, \tag{4.1}$$

where $\vec{V} = (U, V, W)$, each element representing one component of velocity flow along one of the three axes: U is the positive west-east direction on the x axis, V is the positive north-south direction on the y axis and the W is the positive bottom-top direction vertically on the z axis. $\vec{V}_{base} = (U_{base}, V_{base}, W_{base})$ represents the constant values assigned on each component. $\vec{V}_{var} = (U_{var}, V_{var}, W_{var})$ represents the variation values added later.

To test the performance under various weather conditions, we choose 6 different values for base velocity:

1. Group 1: Base velocity is set as $\vec{V}_{base} = (20.0, 20.0, 20.0)$, where all three components of full velocity are assigned the same values. This is the simplest case of our synthetic velocity model.

2. Group 2: Base velocity is set as $\vec{V}_{base} = (20.0, 10.0, 5.0)$. According to the observations on the real data, this is close to the real wind motion in the Great Lakes area.

3. Group 3: Base velocity is set as $\vec{V}_{base} = (5.0, 5.0, 5.0)$. This one is similar to group 1 but has smaller base velocity values. We examine how the retrieval is influenced by the changing of the overall velocity magnitude.

4. Group 4: Base velocity is set to $\vec{V}_{base} = (20.0, 20.0, 5.0)$. This one has very limited motion on the z axis compared to that on the x and y axes. Here we test the influence of a relatively small motion in the z direction on the overall retrieval and how the retrieved vertical velocity W is affected.

5. Group 5: Base velocity is set as $\vec{V}_{base} = (20.0, 5.0, 20.0)$. This is for comparison with the previous group 4. We try to examine what the difference is between the retrievals on different axes.

6. Group 6: Base velocity is set as $\vec{V}_{base} = (5.0, 10.0, 20.0)$. This one, which is the reverse of group 2, furthers our investigation on the differences of retrievals in the z direction compared to that on x and y axes.

From observing historical data, we believe the simulation using group 2 is the most similar to the actual wind motion around the Great Lakes area. The other groups of synthetic data are used to evaluate how the performance would change under potentially extreme conditions. In particular, it helps us understand better how the performance changes as the components on the x, y, and z axes vary. Comparisons between these six groups of data, from which we can explore some characteristics from our 3D velocity recovery algorithm, are given in detail in the next section.

$\vec{V}_{var}$ is introduced to add variation to the synthetic data. The variation is proportional to the base velocity $\vec{V}_{base}$ and can be adjusted with different variation levels, K. Here we use 6 levels of variation in total. As K increases, more variation is added to the data. The highest one is 50% of the base velocity when $K = 5$. The distribution of variation added to each voxel is determined by a *sine* distribution function. The synthetic velocity in the K^{th} variation level can be calculated as in Equation (4.4):

$$U_{var} = U_{base} \cdot \frac{K}{10.0} \cdot sin\left((x - \omega_x) \cdot \frac{PI}{180.0}\right), \tag{4.2}$$

$$V_{var} = V_{base} \cdot \frac{K}{10.0} \cdot sin\left(((y - \omega_y) \cdot \frac{PI}{180.0}\right), \tag{4.3}$$

$$W_{var} = W_{base} \cdot \frac{K}{10.0} \cdot sin\left((z - \omega_z) \cdot \frac{PI}{180.0}\right), \tag{4.4}$$

where the three phase parameters ω_x, ω_y and ω_z add changes according to the precis positions, while variation level K and the base velocity determine the highest variation that could be implemented. It should be noted that variation level $K = 0$ will generate no variation at all, so the synthetic velocity will be constant everywhere. As K increases from 0 to 5, the possible maximum value of variation will change from 0% to 50% of the base velocity $\vec{V}_{base}$, each increment in the K-value representing a 10% increment.

4.1.2 Radial Velocity And Noise Generation

Radial velocity can be considered as the projection of full velocity on the direction of radial beams. The calculation of radial velocity is:

$$V_r = \vec{V} \cdot \hat{r} = (U, V, W) \cdot (r_x, r_y, r_z) = (Ur_x + Vr_y + Wr_z),$$

where $\vec{V}$ represents the full velocity and $\hat{r}$ is the unit vector of the radial direction.

Noise is also added to the experimental procedure as a simulation of the potential pollution to the real data during the radar detection procedure. We add noise to the radial velocity directly. The distribution of random noise is set according to a Gaussian normal distribution. The density function can be described as:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}, \tag{4.5}$$

where μ is the mean value of the distribution and σ is the standard deviation. In our experiment μ is always set to be 0 to get an unbiased noise around 0.0. The value of σ is used to control the distribution shape. Here we use $\sigma = 1.0$. The Gaussian distribution in this experiment is generated using the Teichroew method [50]. (It produces a standard normal distribution $N(0, 1)$, *i.e.* with $\mu = 0$ and $\sigma = 1.0$.) To estimate the error quantitatively, we also define the **Input Error**, E_I, as a ratio of the input noise magnitude to the radial velocity magnitude. In our experiment we have 5 noise levels, L, where the input error E_I is set as 0%, 5%, 10%, 15%, and 20% of the overall radial velocity.

Here we present how to adjust the generated noise to the required noise level. Assume the total number of data points involved in the calculation to be N. All the radial velocities can be treated as components of one big vector, $(\vec{V}_r) = (V_{r_0}, V_{r_1}, V_{r_2}, \ldots, V_{r_N})$. We can calculate the magnitude of this vector as $||\vec{V}_r||_2$ in Equation (4.6). Similarly, the magnitude of all the noise can be calculated as $||\vec{n}||_2$ in Equation (4.7). Divide all the generated Gaussian noise by $||\vec{n}||_2$. We have the $\vec{n}$ normalized, and the new noise $n_i^{'}$ at each data point (x, y, z) can be calculated as:

$$||\vec{V}_r||_2 = \sqrt{V_{r_1}^2 + V_{r_2}^2 + \ldots + V_{r_i}^2 + \ldots + V_{r_N}^2}, \tag{4.6}$$

$$||\vec{n}||_2 = \sqrt{n_1^2 + n_2^2 + \ldots + n_i^2 + \ldots + n_N^2}, \tag{4.7}$$

$$n_i^{'} = \frac{n_i}{||\vec{n}||_2} \cdot E_I \cdot ||\vec{V}_r||_2, \tag{4.8}$$

where n_i is the old noise at any data point (x, y, z), and $n_i^{'}$ is the modified one. E_I is set as 0%, 5%, 10%, 15%, or 20% according to the current noise level L.

4.2 Quantitative Analysis

The quantitative analysis includes output error, E_O, magnitude error, E_M, direction error, E_D and component angular errors, $E_{A_{x_i}}$, $E_{A_{y_i}}$ and $E_{A_{z_i}}$.

4.2.1 Output Error

Output Error, E_O, is used to measure the difference between the retrieved velocity flow and the correct one. The calculation of E_O is similar to the calculation of input error. It can be described as:

$$E_O = \frac{||\vec{V}_c - \vec{V}_e||_2}{||\vec{V}_c||_2} \times 100\% = \frac{\sqrt{\sum_{i=1}^{N} ||\vec{V}_{c_i} - \vec{V}_{e_i}||_2^2}}{\sqrt{\sum_{i=1}^{N} ||\vec{V}_{c_i}||_2^2}} \times 100\%, \tag{4.9}$$

where $\vec{V}_c$ is a vast vector consisting of all the correct full velocities in the dataset, $\vec{V}_c = (\vec{V}_{c_0}, \vec{V}_{c_1}, \dots, \vec{V}_{c_i}, \dots, \vec{V}_{c_N})$. $\vec{V}_e$ consists of all the recovered velocities, $\vec{V}_e = (\vec{V}_{e_0}, \vec{V}_{e_1}, \dots, \vec{V}_{e_i}, \dots, \vec{V}_{e_N})$.

4.2.2 Magnitude Error and Direction Error

Compared to E_O, the magnitude error, E_M, and direction error, E_D report the magnitude and direction deviation between the estimated and correct velocities. These two measurements are first calculated for each data point. Magnitude error E_{M_i} represents the error of the estimated magnitude compared to the correct one at data point i, while the direction error, E_{D_i}, reports the error in the estimated direction as follows:

$$E_{M_i} = \frac{|||\vec{V}_{e_i}||_2 - ||\vec{V}_{c_i}||_2|}{||\vec{V}_{c_i}||_2} \cdot 100\%, \tag{4.10}$$

$$E_{D_i} = \arccos\left(\frac{\vec{V}_{e_i} \cdot \vec{V}_{c_i}}{||\vec{V}_{e_i}||_2 ||\vec{V}_{c_i}||_2}\right) \cdot \frac{180^\circ}{\pi}, \tag{4.11}$$

where $\vec{V}_{e_i}$ is the retrieved velocity at point i, and $\vec{V}_{c_i}$ is the correct one. To summarize the overall performance, we report the average of the magnitude error $\bar{E}_M$ and direction error $\bar{E}_D$, to indicate the performance of the algorithm as a whole. The standard deviation of these average values are also reported as:

$$\bar{\sigma}_M = \sqrt{\frac{\sum_{i=1}^{N}(E_{M_i} - \bar{E}_M)^2}{N-1}}, \tag{4.12}$$

and

$$\bar{\sigma}_D = \sqrt{\frac{\sum_{i=1}^{N}(E_{D_i} - \bar{E}_D)^2}{N-1}}. \tag{4.13}$$

4.2.3 Angular Error on X, Y and Z

The last three measurement metrics are the angular errors on the three components of the full velocity, $E_{A_x}, E_{A_y}, E_{A_z}$. Given the estimated velocity $\vec{V}_e = (V_{ex}, V_{ey}, V_{ez})$ and the correct velocity $\vec{V}_c = (V_{cx}, V_{cy}, V_{cz})$, the component angular error on the x, y, and z axes at data point i can be calculated by Equation (4.14), using modified vectors as $\vec{V}'_{ex} = (V_{ex}, 1)$ and $\vec{V}'_{cx} = (V_{cx}, 1)$. Note that we use $\hat{V}$ to indicate that $\vec{V}$ has been normalized.

$$\begin{aligned} E_{A_{xi}} &= \arccos(\hat{V}'_{c_{x_i}} \cdot \hat{V}'_{e_{x_i}}) \cdot \frac{180^\circ}{\pi}, &(4.14)\\ E_{A_{yi}} &= \arccos(\hat{V}'_{c_{y_i}} \cdot \hat{V}'_{e_{y_i}}) \cdot \frac{180^\circ}{\pi}, &(4.15)\\ E_{A_{zi}} &= \arccos(\hat{V}'_{c_{z_i}} \cdot \hat{V}'_{e_{z_i}}) \cdot \frac{180^\circ}{\pi}. &(4.16) \end{aligned}$$

The component angular errors $E_{A_{x_i}}, E_{A_{y_i}}, E_{A_{z_i}}$ capture both magnitude and direction errors using single numerical value. It successfully avoids the zero division problem inherent in relative error metrics [3]. The angular error is a value varying between $[0, 180]$ degrees (or $[0, \pi]$ radians): the smaller an angular error value is, the better the performance was; an angular error closer to 180 degree means the estimated value is very different from the correct one and therefore very poor. To measure the overall performance, again we present the averaged errors as:

$$\bar{E}_{A_X} = \frac{\sum_{i=1}^{N} E_{A_{X_i}}}{N}, \tag{4.17}$$

$$\bar{E}_{A_Y} = \frac{\sum_{i=1}^{N} E_{A_{Y_i}}}{N}, \tag{4.18}$$

$$\bar{E}_{A_Z} = \frac{\sum_{i=1}^{N} E_{A_{Z_i}}}{N}. \tag{4.19}$$

4.3 Experimental Results Using the Least Squares Method

Here we present the experimental results of the synthetic data experiment using both single radar and dual radar least squares methods. We simulate the same radars, Detroit and Cleveland, that we use in the real datasets. The Doppler radars' positions are shown in Figure 1.5. We assumed the difference between their elevations due to the earth's curvature can be neglected. As shown in the figures discussed below, there is a huge lens-shaped area overlapped by both radars which is the area where the dual algorithm is effective. First we discuss how to decide the parameters in our experiment, then we present the images of correct and recovered full velocity, which are followed by the tables displaying various error metrics.

The selection of various parameters in the optical flow velocity retrieval approaches has been investigated in many previous works such as Chen [6, 14, 15, 13]. Their parameters were chosen through a series of trial and error, and are believed to generate reasonable results. Therefore in our experiment, the common parameters that are shared by our multiple solutions and the original ones are set mainly as the same as their choices with minor changes. Especially in the regularization approach, we introduce one extra relative weight δ_i for each radar i in the multiple radar case. This weight is used to represent the different weights on the radial velocity data from various radars. For now we set them all to be 1.0 in accordance to the original regularization method. However, further investigation that differentiate this value for each radar according to the relative distance from the data point to this radar would be an interesting topic in future research. Table 4.1 lists the values of

Parameter	Value	Meaning
$Thres_{rv}$	1.0	The minimum radial velocity value in use
$Size_m$	$3 \times 3 \times 3$	The matrix size in least squares calculation
τ_1	0.00001	The threshold of the smallest eigenvalue
τ_2	0.01	The threshold of the biggest eigenvalue
$Cond$	100000	The ratio of biggest eigenvalue to the smallest eigenvalue
α	5.0	The weight assigned to smoothness constraint
β	1.0	The weight assigned to least squares constraint
δ	1.0	The weight assigned to radial velocity constraint
Num_{Iter}	150	The maximum number of regularization iteration
$Thres_{Iter}$	0.001	The convergence value

Table 4.1: The values of parameters chosen for 3D full velocity retrieval experiments using dual least squares and regularization methods.

the parameters we use in both our least squares and regularization experiments.

4.3.1 Velocity Flow Images

The velocity flow retrieved from all the six groups of synthetic experiments are displayed in Appendix A. For every group of data, we show the velocity figures for two variation levels: $K = 0$ and $K = 5$. In each variation level, we show the correct synthetic velocity flow and the recovered flow with input error set to be 20% using both the single and dual methods. For example in group 1, Figure A.1 shows the correct velocity in variation level $K = 0$ of group 1. Figure A.2 shows the recovered velocity using single method for $E_I = 20\%$ while Figure A.3 shows the full velocity from the dual method with the same input error. Then Figure A.4, Figure A.5 and Figure A.6 show us the same content with variation level set to $K = 5$.

In each figure, we show 4 pictures: One for the UV velocity field (in the xy plane), and then the x, y and z component velocity fields, U, V, W separately. Taking Figure A.1 for example, Figure A.1a shows the correct UV velocity field. The colours in the background represent the radial velocity that the Doppler radars detected, using the colour magnitude correspondence as in Figure 1.3b. The arrows in this figure represent the full motion we retrieved on the xy plane. The arrow length represents the velocity magnitude while the arrow head indicates the velocity direction. Figure A.1b, Figure A.1c and Figure A.1d show the velocity components on the x, y and z axes, individually. The colours now represent the recovered velocity component's magnitude, $|U|$, $|V|$ or $|W|$, using the same colour magnitude correspondence as in Figure 1.3b. The arrows imposed onto the colour images show the motion's magnitude and also the direction. Since they only show the velocity values on one component, their directions can only be positive or negative. Thus in Figure A.1b the arrows are all heading to the positive (east) or negative (west) direction of the x axis, while in Figure A.1c they are positive (south) or negative (north) on the y axis. In order to show the velocity component W in the 2D image, we set the vector direction of 45° counterclockwise from x positive direction as the positive direction of the z axis as shown in Figure A.1d. In order to distinguish the arrows of the Detroit radar data from that of the Cleveland radar, the arrows from the Cleveland radar results are shown in dashed lines.

It should also be noted that the data we present here are all from the first elevation from their datasets. Our program is functionally able to display various elevations separately. But as we mentioned in Chapter 1, the NEXRAD *II* radar data has very limited capability in the upward direction due to its fewer elevations compared to the the NEXRAD *I* radar data. Besides, we observe that in the real data the sizes of coverage areas are also decreased dramatically in the higher elevations. Therefore we mainly focus on displaying the first elevation results in both the synthetic and real experiments. The data in the other elevations show similar results while their coverage areas are much smaller. Their results are also used in the following error analysis section.

In general, we find both the single and dual methods can generate reasonable results compared with the correct synthetic velocity. When the variation level was set to $K = 0$, the retrieved velocity field tends to be uniform, while with $K = 5$, there are many variations fluctuating in the velocity field. The results displayed here using $L = 4$ ($E_I = 20\%$) present the performance in the highest level of noise in our experiment. The results for the other input noise levels will generate better results compared to it. They are not shown due to length limitations, but the measurement of their performance can be found in the following error analysis section. More importantly, the velocity flow retrieved using the dual radar method is more similar to the true velocity in the common coverage area than the velocity recovered by the single radar method, indicating that the dual velocity recovery method performs better.

4.3.2 Error Analysis

Following the velocity flow images, in each group of data, we present the error metrics of synthetic data under various variational levels and noise levels using both the single and dual least squares methods.

For example in Group 1, Figure A.7 and A.7 show the error metrics when the base velocity was set to $(20.0, 20.0, 20.0)$. Figures A.7a, A.7c, and A.7e describe the output error, average magnitude error and average direction error in the single case, while Figures A.7b, A.7d and A.7f show the same metrics in the dual case. Figures A.7g, A.7i and A.7k show the angular errors for the three velocity components in the single case while Figure A.7h, Figure A.7j and Figure A.7l show the same in the dual case. In each figure, curves with different colours represent the performances of different variation levels with the input error changes through 0%, 5%, 10%, 15%, and 20%. The blue curve shows the performance of variation level $K = 0$, while red for $K = 1$, green for $K = 2$, purple for $K = 3$, cyan for $K = 5$ and orange for $K = 5$. It is obvious that in all the figures, higher variation levels or higher input errors cause worse performance.

In all these figures, the dual radar method always gives better results than the single case, no matter what the base velocity, variation level or input error is. Take Figure A.7

for example, the output error has dropped about 10% from the single case to the dual one. The improvements on magnitude error and direction error are even larger: The metrics in the dual case has decreased to around half of that in the single case. In Figure A.7 the improvements on angular errors are also obvious.

From Figure A.7, Figure A.14, Figure A.21, Figure A.28, Figures A.35 and Figure A.42, we found that the output error, magnitude error and direction error share a very similar performance for all six synthetic data groups. It seems that no matter how we change the base velocity, such as decreasing the base velocity values from $(20.0, 20.0, 20.0)$ to $(5.0, 5.0, 5.0)$, or changing the values on different components from $(20.0, 10.0, 5.0)$ to $(5.0, 10.0, 20.0)$, these three error metrics are always kept in the same range. However, it can be noted that the angular errors on the three components have shown large differences for various groups of synthetic data.

Take group 1 and group 2 synthetic datasets, for example. As shown in Figure A.7 and Figure A.14, the angular errors on the x and y axes are roughly the same, but the error on the z axis in group 2 is almost three times worse as that for group 1. This phenomena indicates that the changes of base velocity on the three velocity components have influenced the precision of our recovery on the three components. Furthermore, comparing the results of group 1 with the group 3, where the base velocity has decreased from $(20.0, 20.0, 20.0)$ to $(5.0, 5.0, 5.0)$, we found no remarkable change on the three angular errors. Thus we deduce that the overall magnitude of base velocity will not effect the recovery very much. The performance changes are caused more possibly by the changes of distribution on the three velocity components.

In order to verify our deduction, look at group 4 where the value on the z axis is very small. As shown in Figure A.28, the angular errors on the x and y axes are similar to group 1 or group 2. However, the angular error on z in group 4 shows the worst performance among all six groups, almost 40° at its maximum. It indicates the recovery on one velocity component will be less successful if the value of the base velocity assigned on this specific component is relatively smaller than on the others. Similar phenomenon can be found in group 5, where a smaller base velocity value is assigned on the V component. Comparing the angular error on the y axis in Figure A.35 with others, it is obvious that in group 5 of the synthetic data, the angular error on the y axis is much higher than on the others. To explain this phenomena, recall in Equation 4.5, where radial velocity is generated by the sum of the projections of all three full velocity components. If one component has a relatively smaller value than the other two, its influence on the radial velocity will also be relatively smaller, which causes the recovery on this component to be more difficult.

Then we compare the results from group 6 shown in Figure A.42 with the flow of group 2 in Figure A.14. The velocity base we set for them are the same except that the values on x and z are switched. As we expect, the angular error on the x axis in group 6 is very high compared to that of group 2, while the angular error on the z axis has dropped remarkably from the previous one up to 30° to the current only 1°. This verifies our previous deduction that a smaller value on one component will cause trouble in the velocity component recovery in this direction.

Besides, comparing Group 6 of the synthetic data with group 2, the highest angular error we get on the x axis is only 10° as shown in Figure A.42g, while that value on z axis is around 35° in Figure A.14k. Furthermore, in the equal-base test of group 1, where all three velocity components have exactly the same magnitude bases, the performance on the z axis is also much worse than that on the x or y axes as shown in Figure A.7. All these results indicate that the retrieval in the z direction is much harder than on the horizontal x and y directions using the least squares method. This is due to the Doppler radar's intrinsic Aperture Problem [3] on the vertically bottom up direction.

4.4 Experimental Results Using the Regularization Method

Here we present the experimental results with the synthetic data using both single and dual radar regularization methods. We use the same locations of the radars, Detroit and Cleveland, as in the least squares case. First, the images of correct and recovered full velocity are presented to exhibit the recovery in each case. Then the various error metrics are shown in separate figures.

4.4.1 Velocity Flow Images

The figures of the velocity flow retrieved from all six groups of synthetic experiments are presented in the same order as in the least squares case in Appendix B. For every group of data, we show the velocity figures for two variation levels: $K = 0$ and $K = 5$. In each variation level, we show the correct synthetic velocity flow and the recovered flow with input error set to be 20% using both the single and dual methods. For example, Figure B.1 shows the correct velocity in variation level $K = 0$ of group 1. Figure B.2 shows the recovered velocity using single radar method for $E_I = 20\%$ while Figure B.3 shows the full velocity from the dual radar method with the same input error. Then Figure B.4, Figure B.5 and Figure B.6 start to show us the same thing again for the variation level $L = 5$, and the next figures are for group 2.

The velocity flow figures show lots of similarities as in the least squares case. Generally in every group of data both the single and dual regularization methods can generate reasonable results compared with the correct synthetic velocity. When the variation level was set to $K = 0$, the velocity field is constant everywhere, and there are many variations fluctuating in the velocity data when K increases. Different from the least squares scenario, the performance with the $L = 4$ ($E_I = 20\%$) looks very smooth in this regularization case. After adopting the dual method, the recovered velocity from two radars are more similar in the common coverage area than in the single case. The recovery become a little bit more difficult and may contain errors in the high variation cases. Compared with the results from least squares method, the velocity flow from regularization method appear to be much more smooth and less noisy. The results are also much closer to the correct velocity flow. It indicates that the recovery using regularization methods tend to be more successful than the least squares ones. To evaluate the performance of our two regularization methods, we presented the quantitative error analysis in the next section.

4.4.2 Error Analysis

As we did in the least squares scenario, following the velocity flow images, in each group of data we also present the error metrics of synthetic data under various variational levels and noise levels using both the single and dual regularization methods. For example, Figure B.7 and B.7 show the error metrics when the base velocity was set to $(20.0, 20.0, 20.0)$. Figures B.7a, B.7c, and B.7e describe the output error, average magnitude error and average direction error in the single case, while Figures B.7b, B.7d and B.7f show the same in the dual case. Figures B.7g, B.7i and B.7k show the angular errors for the three velocity components in the single case while Figure B.7h, Figure B.7j and Figure B.7l show the same in dual case. In each figure, curves with different colours represent the performances of different variation levels with the input error changes through 0%, 5%, 10%, 15%, and 20%. The blue curve shows the performance of variation level $K = 0$, while red for $K = 1$, green for $K = 2$, purple for $K = 3$, cyan for $K = 5$ and orange for $K = 5$. It is obvious that in all the figures, higher variation levels or higher input errors always cause worse performance (more errors).

First of all, comparing the results of regularization method with the least squares one, it is clear that in all six groups of synthetic experiment the regularization method performs better than the least squares method. For example, in the first group of synthetic data, the output error, magnitude error and direction error in Figure B.7 and the angular errors on three directions in Figure B.7 are all smaller than those values in Figures A.7 and A.7, especially in the high noise levels. The output error in the single regularization case in Figure B.7a is only half of that in least squares case shown in Figure A.7a; while the output error in the dual case is dropped from 35% in least squares results in Figure A.7b to about only 20% in the regularization result in Figure B.7b. The reason why dual regularization results cannot improve as much as the single one is because the dual least squares method has already enhance the performance compared with the single case. The other metrics show similar trends. Besides, comparing the dual regularization results with the single regularization results, we can also find that the dual methods usually generates better results than the single ones. But the overall enhancement of performance from single regularization method to the dual regularization method is not so impressive as in the least squares case.

Second, compared with the least squares case shown in Figures A.7k and A.7l, the angular error on z axis in the regularization case has decreased remarkably as shown in Figures B.7k and B.7l. Other than the relatively higher errors in the least squares case, the angular error on z axis is now kept in the same range as the angular errors on x and y axes. It is more obvious when the base velocity is set to a smaller value on the z axis as in group 2 and group 4, where the base velocity is set to $(20.0, 10.0, 5.0)$ and $(20.0, 20.0, 5.0)$ respectively. As shown in Figures A.14k and A.14l, the angular error on z axis using least squares method is about 30° at maximum, much higher than the errors on x or y axes. But the velocity flow generated by regularization method has an angular error on z axis as only 4° at the most, as shown in Figures B.14k, and B.14l, which is in a comparable range as the results on the other two components. Similar changes can also be observed clearly in group 4's results, where the angular error on z axis has declined intensely from almost 40° using single least squares in Figure A.28k to only 6° using single regularization in Figure B.28k. It indicates that the regularization method brings huge benefits in reducing the bias in the recovery on the z direction. This is one of the major advantages of applying regularization method.

Third, compared with the curves from least squares charts, the performance curves using the regularization method tend to be more "flat". As shown in Figure B.7 and Figure B.7, the differences between the performances of error metrics in various noise levels are very trivial. As the noise levels, L, increases, there are more input error ϕ_I added, but the error metrics didn't change as much as in the least squares case. It seems the recovery using regularization method is more stable, less vulnerable to any change in the added noise. The only factor that influences its performance are the base velocity we use and the variation level, K, that we set up. Once these two factors are chosen, no matter how the noise level varies, the performance of regularization method will remain roughly the same as in the first noise level where $\phi_I = 0\%$ with no input error. The stability of regularization recovery through different noise levels indicates that the regularization method is less sensitive to any potential noise than the least squares one. Therefore, we can conclude that the global regularization process tends to produce a recovery with less influence of input error, giving us a much more stable and robust results than the least squares method.

There are still very few curves that change dramatically in the charts, such as the angular errors on z axis in group 2 as shown in Figures B.14k and B.14l, and that in group 4 in Figures B.28k and B.28l. The angular errors there still increase intensely when more noise is added. It indicates that the recoveries in these cases are not as stable as in other groups. We attribute the instability of performance in these two groups of data to the W value of base velocity we chose in group 2, $(20.0, 10.0, 5.0)$ and group 4, $(20.0, 20.0, 5.0)$. The values in z direction in these two cases are all much smaller than those in the other two directions. As we discussed in the least squares section, a relative smaller component compared to the other directions will always get worse performance in the recovery. However, using these regularization methods, the absolute values of angular errors on z direction have already been suppressed into a much smaller range, compared to the results from least squares method.

In order to investigate this instability phenomena in details, we can compare the results from group 2 with that from group 6. In group 6 the base velocity is set to $(5.0, 10.0, 20.0)$, which is the reverse of group 2, thus the value on x axis is the smallest one. As expected, we can find that the recovery on the x axis become less successful now. As shown in Figure B.42g, the angular error on x axis using single method also changes in a wider range in the single case, and the value is about 3 times bigger than that on y or z axes; while in Figure B.42h, the angular error on x axis using dual method tends to be less changeable but is still higher compared to that on the other directions. Similar results can be found by comparing the results from group 4 and group 5, where the base velocity has been set to $(20.0, 20.0, 5.0)$ and $(20.0, 5.0, 20.0)$ respectively. The direction that has the smallest base velocity value always has the biggest angular error, as shown in Figure B.28 and Figure B.35. Nevertheless, the performance on the x axis in group 6 in Figure B.42 and on the y

axis in group 5 in Figure B.35 are not so unstable as on the z axis shown in Figures B.14 or B.28. Again we deduce this problem to the Doppler radar's intrinsic Aperture Problem [3] that makes the retrieval on z axis much harder, as we discussed in the least squares section.

4.5 Real Data Results

Here we present the recovered velocity flow from real data, so the performance in a real environment can be judged qualitatively. Our raw data are collected from the Detroit and Cleveland radar sites of the NCDC (National Climate Data Center) network. The data from the Detroit site were obtained at 13:59:12 of August 20^{th}, 2007, and the data from Cleveland were at 14:00:30 of the same day. The positions of radars can be found in Figure 1.5. It is clear in the real datasets that the Detroit radar site covers a smaller area than the Cleveland radar at that moment. The display scheme is the same as in the synthetic dataset experiment. In each figure, we show 4 pictures: One for the UV velocity field (in the xy plane), and then the x, y and z component velocity fields, U, V, W, separately. In this real dataset experiment, we adopt the same values for all the parameters as in the synthetic experiment, except that we only process the radial velocity values higher than 1.0 km/hr in order to avoid any unnecessary unbiased noise in the raw data.

4.5.1 Results Using the Least Squares Method

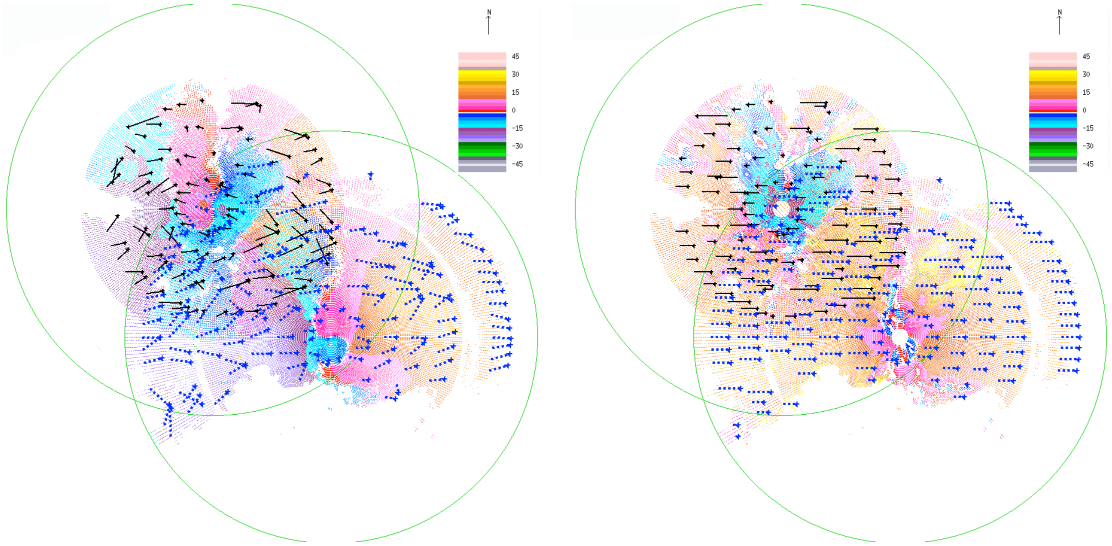
Figures 4.1a and 4.2a show the recovered UV flow using the single and dual radar least squares methods. The background colours show the radial velocity that our recovery is based on. The arrows show both the magnitudes and directions of the UV components of the full velocity. It is obvious that the results from the dual case show more similarities than the single one in the common overlapping area. The velocity flow tends to move in a common direction with similar magnitudes in Figure 4.2a. However it can also be observed that there are some abnormal large velocity vectors recovered in the dual case (Figure 4.2a). This is caused by the fact that the radial velocities provided by the two radars are conflicting with each other. Their motions have nothing in common, so the results generated by their combination are very unstable.

The results of three velocity components using the single radar method as shown in Figures 4.1b, 4.1c and 4.1d and using the dual radar method in Figures 4.2b, 4.2c and 4.2d show similar trends as the UV flow. The retrieved velocity components share more similarities in the common overlapped area in the dual case than in the single case. However there are occasionally some abnormal fluctuations in the magnitudes observed in the dual case. This implies the drawback of the least squares method is that it is limited in its ability to filter the abnormal noise from the results.

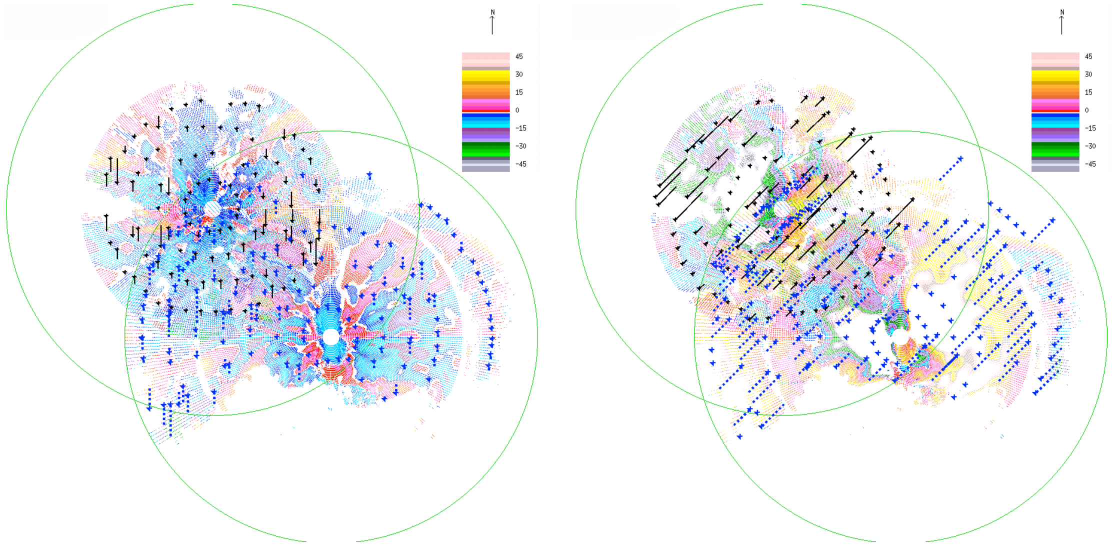
4.5.2 Results Using the Regularization Method

Figures 4.3a and 4.4a show the recovered UV flow using single and dual radar regularization methods. Again, compared with the single radar results shown in Figure 4.3a, the results from the dual radar case in Figure 4.4a display more similarities to the velocity motion. The velocity flow on the UV plane shows a consistent movement from west to east with constant magnitude. Compared with the dual least squares results shown in Figure 4.2a, the previous unstable vectors have been filtered out successfully, so the regularization method is able to provide much smoother results.

The results of the three velocity components using the single radar regularization method is shown in Figures 4.3b, 4.3c and 4.1d, and the dual radar results are in Figures 4.4b, 4.4c and 4.4d. The three components' results show similar trends as in the UV flow. Generally, both the single and dual radar regularization methods provide more consistent results than both of the least squares methods. Compared with the results from the single radar regularization method, the results from the dual one are more stable. The background colours representing the magnitude changes vary less frequently and have fewer abrupt fluctuations, especially in the W component as shown in Figure 4.4d. So overall the dual radar regularization method offers more reliable results than the single radar case or the previous least squares method. This conclusion is also in accordance with our synthetic experiments



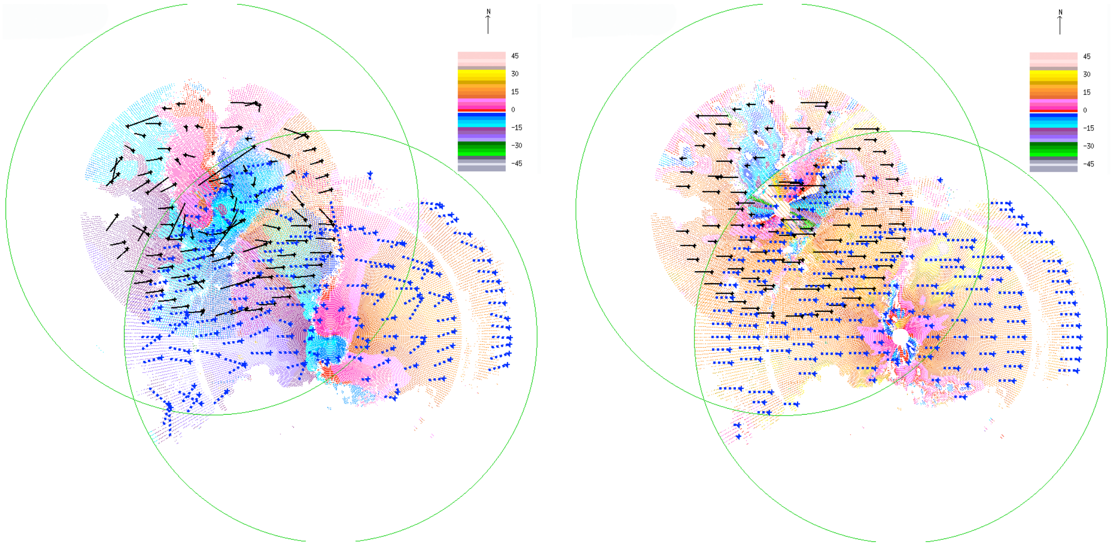
(a) UV (b) U



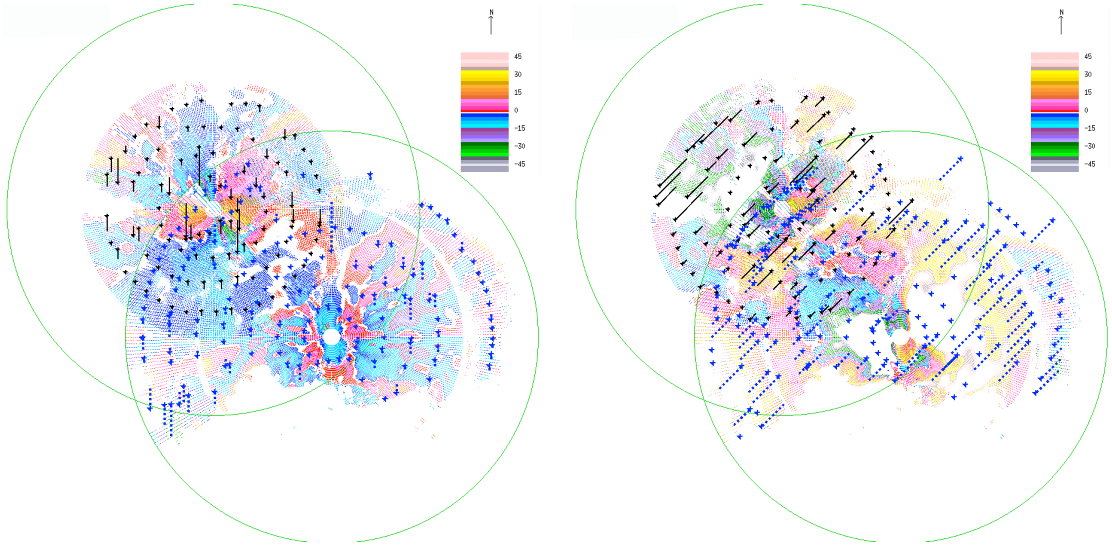
(a) V (b) W

Figure 4.1: 3D full Velocity Flow using single Least Squares methods on real data at 13:59:12 of August 20^{th}, 2007 at Detroit and at 14:00:30 of August 20^{th}, 2007 at Cleveland. (a) the UV flow, (b) the U component, (c) the V component and (d) the W component.

addressed in the previous section.



(a) UV (b) U

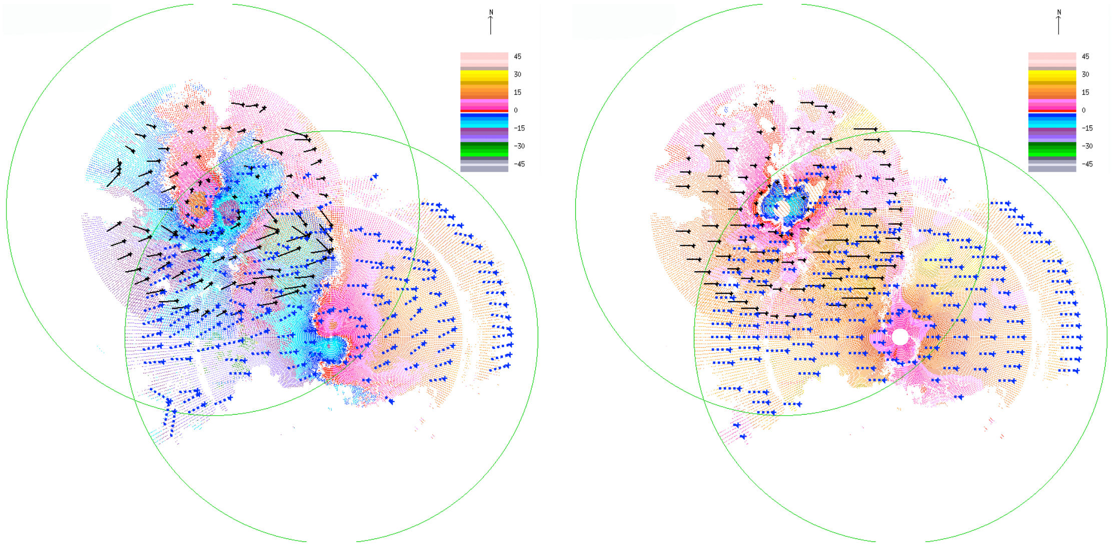


(a) V (b) W

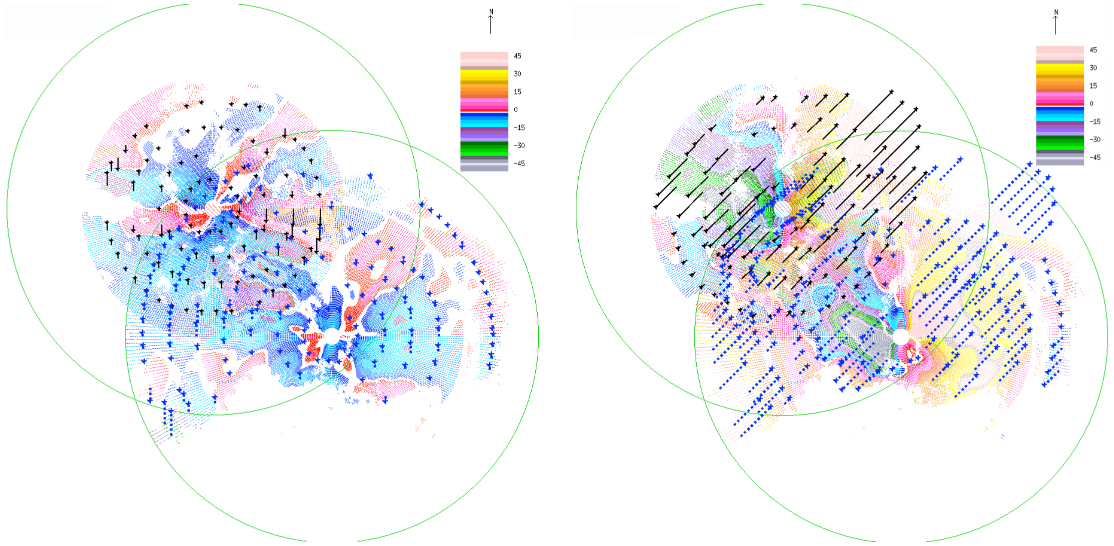
Figure 4.2: 3D full Velocity Flow using dual Least Squares methods on real data at 13:59:12 of August 20^{th}, 2007 at Detroit and at 14:00:30 of August 20^{th}, 2007 at Cleveland. (a) the UV flow, (b) the U component, (c) the V component and (d) the W component.

4.6 Experiment Discussions

In order to verify the reliability and accuracy of our velocity retrieval methods, we have designed a synthetic experiment. We have tested 6 groups of synthetic data, together with



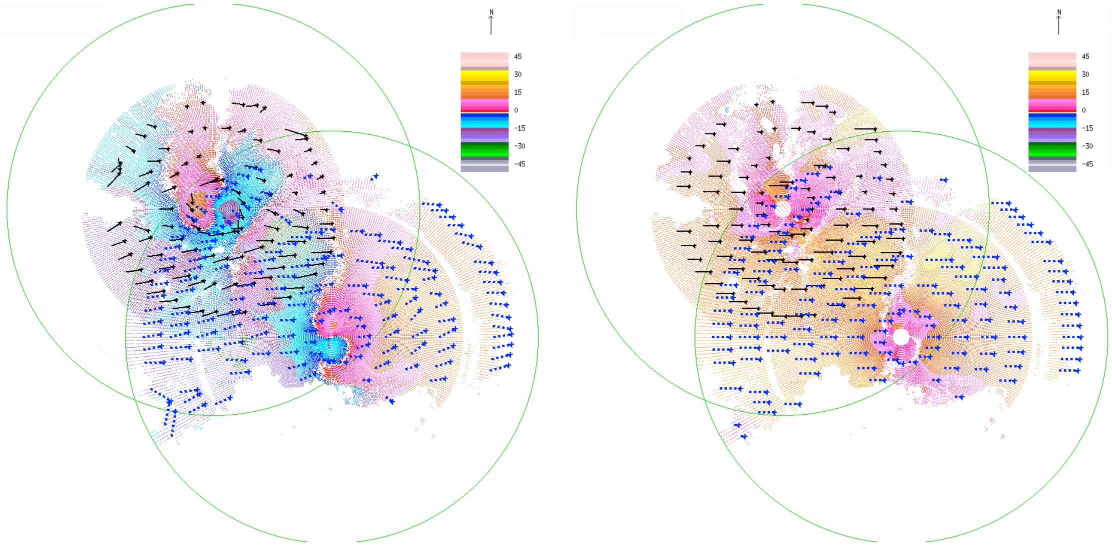
(a) UV (b) U



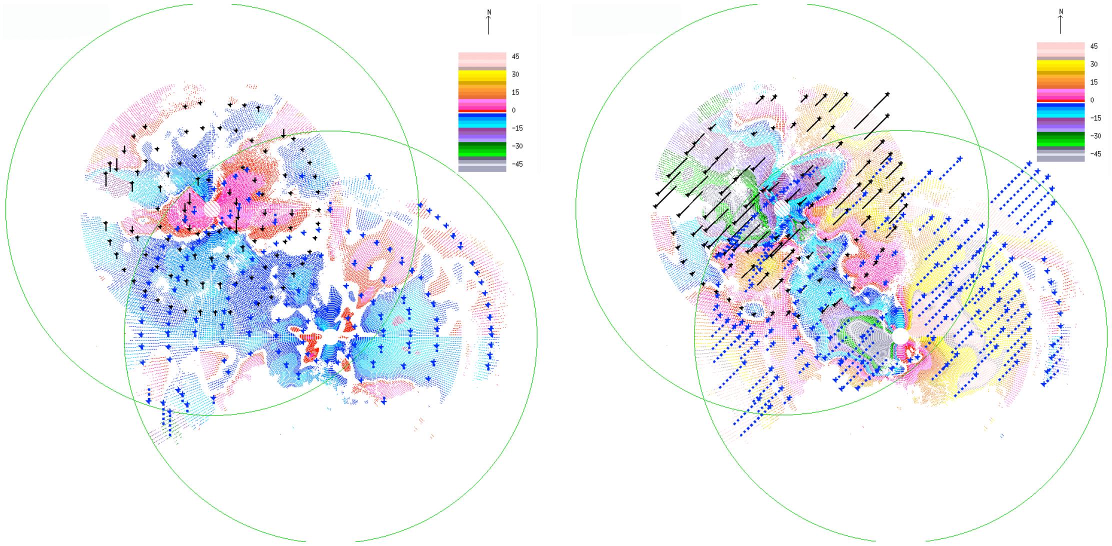
(a) V (b) W

Figure 4.3: 3D full Velocity Flow using single Regularization methods on real data at 13:59:12 of August 20^{th}, 2007 at Detroit and at 14:00:30 of August 20^{th}, 2007 at Cleveland. (a) the UV flow, (b) the U component, (c) the V component and (d) the W component.

6 levels of variation and 5 levels of noises. We also propose several error metrics such as output error, magnitude error, direction error and angular errors to help evaluate the



(a) UV (b) U



(a) V (b) W

Figure 4.4: 3D full Velocity Flow using dual Regularization methods on real data at 13:59:12 of August 20th, 2007 at Detroit and at 14:00:30 of August 20th, 2007 at Cleveland. (a) the UV flow, (b) the U component, (c) the V component and (d) the W component.

performance of our recovery. Qualitative evaluations of the performance with real data are also presented.

In the least squares case, by comparing the results of the dual radar least squares method with the results in the single radar case, we find that the dual radar method works much better than the single radar one. This improvement is reliable, not suffering much from the changes of the preset base velocity or the potential variations and noise. Based on this, the results using both single and dual radar regularization methods can smooth the results from the least squares method, making it less sensitive to the system input noise, and decreasing the errors in the velocity's three components. The dual radar regularization method presents a better performance than the single radar regularization one, though this improvement is not so strong as in the least squares case. By testing different groups of synthetic data, it is safe to conclude that the improvement of the regularization method is reliable. Through various tests with different base velocities or variation levels, the regularization method is capable of providing consistently satisfactory performance with very little influence from the input noise. It also reduces the angular errors in three directions, especially on the z axis, where the error is now comparable to that on the other two axes. Therefore, the regularization method surpasses the others for our full velocity retrieval process in the multiple Doppler radars case. Our real data experiment presents similar results as for the synthetic one.

However, it is noteworthy that varying the values of the base velocity in the three components still causes large fluctuations in the performances of these components. It is due to the inevitable Aperture Problem [3]. The use of wind profiler data [34] is suggested in the next chapter as another constraint to be implemented into our full velocity recovery scheme to help solve this issue.

Chapter 5

Refinement of 3D Full Velocity Using Windprofiler Radar

In this chapter we discuss about the cooperation of windprofiler with traditional Doppler radars. As we previously addressed in Chapter 2, windprofiler has higher accuracy especially on upward direction compared to the Doppler radars. By introducing it into our 3D full velocity retrieval procedure, we expect an enhancement of accuracy in velocity retrieval. This refinement is based on Horn and Schunck [37]'s global regularization solution which is naturally open to introducing additional different constraints. Besides, we also test the same method using radiosonde data obtained during the experiment at Harrow station in June and July 2007. This extended solution presents similar results as windprofiler data, indicating that the generalization of our solution is promising.

5.1 The Windprofiler Data Structure

Different from the Doppler radar, the windprofiler radar only detects one dimension of data at one time. Then the data obtained from different directions are are collected to generate the full velocity. A windprofiler can only cover a small range of area compared to the Doppler radar. It records the wind's motion at different height levels in the upwards direction, and saves the data as a list of arrays. Each array is recorded for one time period, the interval between arrays varies from 20 minutes to one hour. In our data, each array has 30 data points along the upwards direction. The resolution of the data points is $500m$, so the top height of the windprofiler data at $15km$.

Currently, we collect data from Doppler and windprofiler radars around the Great Lakes area. The distribution of Doppler radars and windprofiler radars in this area can be found in Figure 1.5 of Chapter 1. The NEXRAD II data are acquired from the Detroit and Cleveland radars via the NCDC (National Climate Data Center) network in the US. The windprofiler data is obtained from Canadian windprofiler radars at Harrow and Walsingham. We limit our discussion to the data from Detroit and Harrow for now but it can be further extended to the other radars without any difficulty. The location of the Detroit Doppler radar is at (42.695°N, 83.467°W) while the Harrow windprofiler radar is at (42.040°N, 82.892°W). The distance between them is about $86.94km$.

Figure 5.1 shows a Doppler radar image at elevation 0 acquired on August 17^{th} by the Detroit radar. The coordinate system used in the 3D velocity representation is the same as before, shown in Figure 1.3a. We also use the same colour magnitude correspondence as in Figure 1.3b to represent velocity flow. The black dot in the lower right corner centered in a blue circle indicates the position of the windprofiler at Harrow relative to the Doppler radar in the center. To focus the reader's attention on the windprofiler area, we have cropped subsequent colour velocity fields to contain just the windprofiler radar area with just a bit of the Doppler radar data showing.

5.2 The Refined Optical Flow Calculation

One significant advantage of the regularization approach to computing optical flow is that it is open to adding many additional constraints. After adding a least squares constraint as in the previous chapter, here we also incorporate the windprofiler estimates into our process. We refer to the velocity flow computed by the original regularization method as

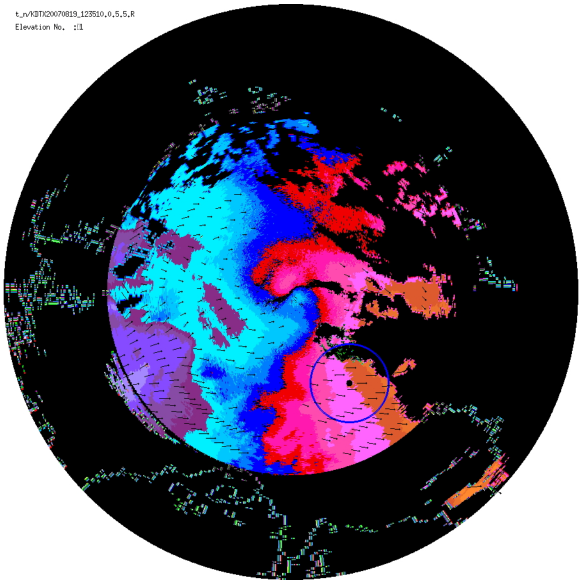


Figure 5.1: An example of coloured Doppler radial velocity data (from elevation 0, August 19^{th}, 2007 Detroit Doppler data) with the location of the Harrow wind profiler and its area of influence indicated.

the *unrefined* optical flow, and call the velocity flow retrieved by integrating windprofiler data with the Doppler data the *refined* optical flow.

We add a fourth consistency constraint to the energy function for windprofiler velocity $\vec{V}_{wp}$. We assume the windprofiler radar has observed n points along its height axis, $\vec{V}_{wp_i}$, $i = 0 \ldots n-1$ at positions $x_{wp_i}, y_{wp_i}, z_{wp_i}$. All of these data points influence the full velocities we try to retrieve in the nearby area of the wind profiler radar. However, the influence should decrease as the distance increases between a Doppler radar voxel where we are computing the 3D velocity and these observation points. Our minimization terms become:

$$\begin{aligned}
&\int\int\int \underbrace{(\vec{V}\cdot\hat{r} - V_r)^2}_{\textit{Radial Velocity Constraint}} + \\
&\alpha^2 \underbrace{(U_x^2 + U_y^2 + U_z^2 + V_x^2 + V_y^2 + V_z^2 + W_x^2 + W_y^2 + W_z^2)}_{\textit{Velocity Smoothness Constraint}} + \\
&\beta^2 \underbrace{((U - U_{ls})^2 + (V - V_{ls})^2 + (W - W_{ls})^2)}_{\textit{Least Squares Velocity Consistency Constraint}} + \\
&\sum_{i=1}^{n} \gamma_i^2 \underbrace{((U - U_{wp_i})^2 + (V - V_{wp_i})^2 + (W - W_{wp_i})^2)}_{\textit{Windprofiler Velocity Consistency Constraint}} \partial x \partial y \partial z,
\end{aligned} \tag{5.1}$$

where the γ_i values are the Lagrange multipliers for this additional constraint. The value of γ_i at each voxel is calculated from a 3D Gaussian function based on the distance between it and the location of the i^{th} point of the windprofiler radar:

$$\gamma_i = \frac{\Gamma}{(2\pi)^{\frac{3}{2}} \prod_{k=1}^{3} \sigma_k} e^{-\left(\frac{(x-x_{wp_i})^2}{2\sigma_1^2} + \frac{(y-y_{wp_i})^2}{2\sigma_2^2} + \frac{(z-z_{wp_i})^2}{2\sigma_3^2}\right)}. \tag{5.2}$$

$\sigma_1, \sigma_2, \sigma_3$ are the three standard deviations that specify the shape of the 3D Gaussian distribution according to the distance in each direction, while Γ is a preset constant value as experimentally determined using the synthetic data. For our experiments, σ_1 and σ_2 have the value 20.0, reflecting the large x and y range of values and $\sigma_3 = 0.4$, reflecting the much smaller range in the z values. For the first iteration, $\vec{V}^0$ is set to zero. After a number of iterations, when the difference between the k^{th} and $(k+1)^{th}$ velocities becomes less than a preset threshold τ, the final velocity field is found.

5.3 Experimental Results

We first present results using real data obtained from the Detroit Doppler radar and the wind profiler radar at Harrow. There are many changes to the Doppler-only wind motion, which indicates that the new wind profiler constraint does modify our original unrefined scheme. However, since we do not know the correct 3D flow in the real data case, we can only make a qualitative analysis. Therefore, we again test our refined method using synthetic Doppler and windprofiler data to evaluate the performance quantitatively. Further, we discuss the effect on the optical flow for different parameter values in the regularization functional.

5.3.1 Real Data Experiment

This section presents the results from the real radar data. The windprofiler radar is at Harrow, which is southeast of the Doppler radar at Detroit. The main wind motion in this area is from West to East.

Figures 5.2a and 5.2e show the unrefined and refined UV optical flow fields for the Detroit Doppler data at 12:35:10 on August 19^{th}, 2007. We used $\sigma_3 = 1.0$ and $\Gamma = 1000$ in our computation. It is clear from these flow field images that using the refined method changed the flow field around the windprofiler radar significantly. For example, the direction of velocity was significantly changed from the unrefined flow that goes to the east in Figure 5.2a to refined flow that goes to the south in Figure 5.2e.

Figures 5.2b and 5.2f show the coloured U component optical flows. The orange colour indicates a positive U component magnitude of about 15 m/s in the blue circle (see the colour-magnitude correspondence map in Figure 1.3b). We see that the directions of the U component remains the same, but its magnitude becomes slightly smaller near the outer area of the blue circle, which indicates that at this position the windprofiler detects a smaller motion there. These images indicates that the component velocity along the x axis of the Doppler radar is relatively unaffected by adding windprofiler data.

The retrieved V component flow along the y direction shows much smaller magnitudes of V with different directions (red magnitudes range from away 0-5 while blue magnitudes range from towards 0-5) in the unrefined flow in Figure 5.2c. However in the refined component flow in Figure 5.2g the colour pattern has been replaced by a significantly enhanced magnitude in the positive direction (red). The magnitude of this V velocity component is much smaller than the magnitude of the U velocity component and so the overall velocity is relatively unaffected by V. The retrieval of the V component of velocity is significantly

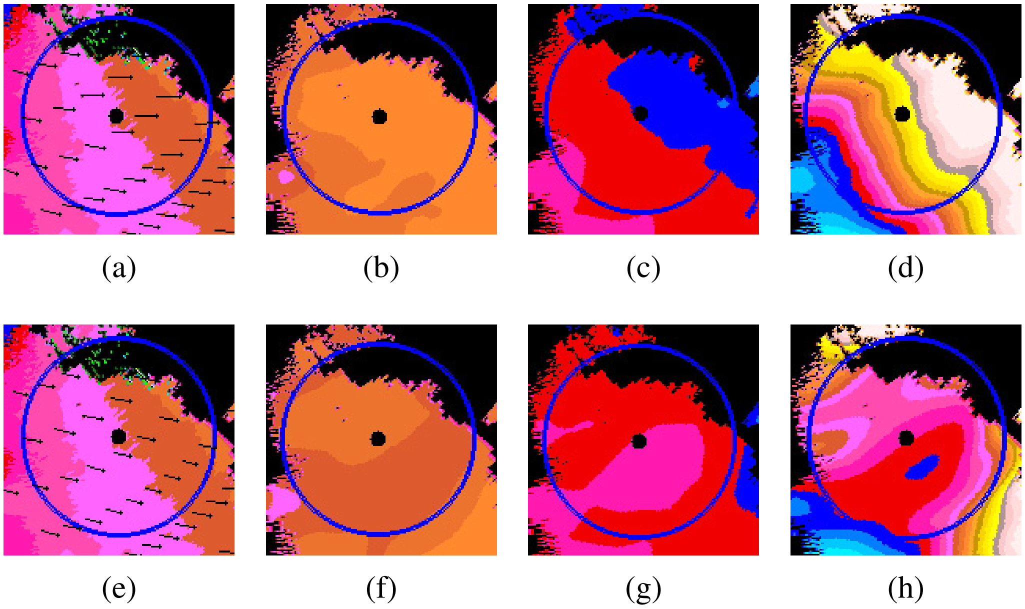


Figure 5.2: Full velocity retrieved from the Detroit Doppler (and the Harrow windprofiler) data on August 19^{th}, 2007 at 12:35:10: (a) the unrefined UV optical flow, (b) the U component of the unrefined optical flow, (c) the V component of the unrefined optical flow, (d) the W component of the unrefined optical flow and (e) the refined UV optical flow, (f) the U component of the refined optical flow, (g) the V component of the refined optical flow, (h) the W component of the refined optical flow.

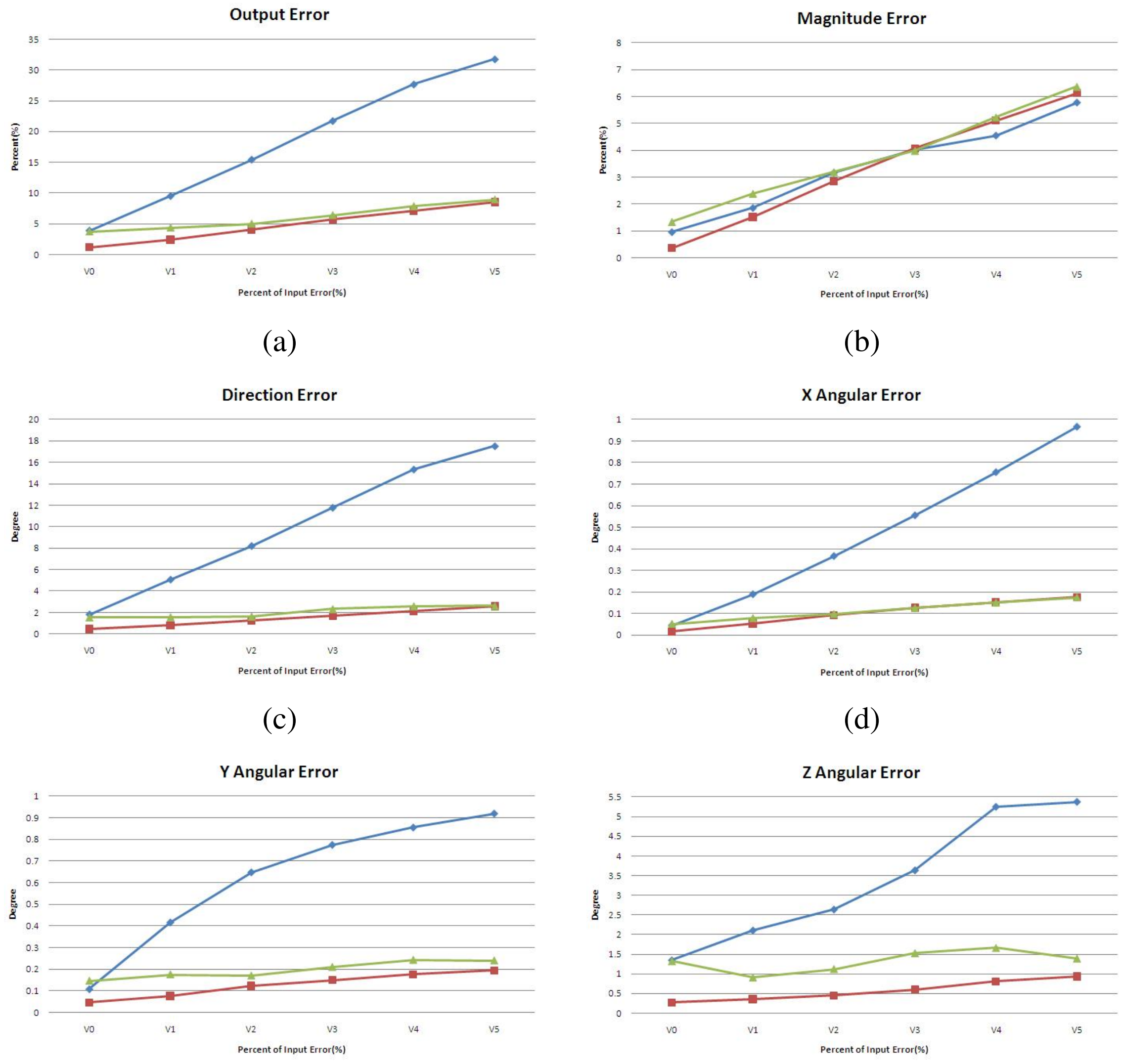
changed by the windprofiler data because the windprofiler radar at Harrow is south of the Detroit Doppler radar. Since the main wind motion in this area is from West to East (the U component velocity is dominant), it is hard to retrieve the V velocity component using the Doppler radar only because this measurement direction is nearly orthogonal to the velocity direction (a manifestation of the aperture problem mentioned before). In this case, we believe the velocity in the Y direction measured by the windprofiler radar is more accurate and should dominate. Thus, integrating the windprofiler data in our optical flow calculation led to an improvement in the calculation of the V component of the optical flow in this real Doppler dataset.

As expected, we see a significant change in the recovery of W in the z direction. In the unrefined W component flow shown in Figure 5.2d, there is a large area where these component velocities have reached their maximum value. This shows that retrieval of the W components is very poor due to the aperture problem. We can see that the refined component flow inside the circle shown in Figure 5.2h is very different and more reasonable than the surrounding Doppler flow outside the circle or inside the unrefined circle in Figure 5.2d. Now we observe a small upward velocity component in this area (red), but with a small downward component (blue) in a small enclosed area near but below the circle's center. This may be due to the fact that the windprofiler constraint uses many different velocity values for each Doppler voxel, according to the voxel's actual height.

5.3.2 Synthetic Data Experiment

We use the NEXRAD II Doppler data structure used at the Detroit radar site and the windprofiler data structure used at Harrow. The synthetic experiment design is the same as the one in Chapter 4. We choose only the synthetic data group 2 for this experiment since $\vec{V}_{base} = (20.0, 10.0, 5.0)$ is believed to be the one closest to the real wind motion around this area. Therefore, we can roughly evaluate the improvement that a wind profiler radar can cause.

To evaluate the influence of noise in the wind profiler data, we use two different noise levels for the wind profiler: one has 0% input error and the other one has 10% input noise. Figure 5.3 shows the average error metrics for 20% input error in the Doppler radar data with the two fixed windprofiler input errors of 0% and 10% while K varies from 0 to 5. The blue curves in each figure show the performance of the unrefined retrieval, while the red curves show the performance of the refined method with 0% input windprofiler error and the green curves show the performance of the refined method with 10% wind profiler input error. It is obvious that the refined method always gives much better performance than the unrefined one, except for the magnitude error, where the performance is roughly the same (large sinusoidal variation in the data causes retrieval problems independent of noise). It indicates that the refined method using wind profiler has a greater influence on the direction



(a) (b)

(c) (d)

(e) (f)

Figure 5.3: Error results of the synthetic data for all K values: (a) the average output error, (b) the average magnitude error, (c) the average direction error, (d) the average x angular error, (e) the average y angular error and (f) the average z angular error. The blue lines show the performance of the unrefined retrieval, the red lines show the performance of the refined method with 0% input windprofiler error and the green lines show the performance of the refined method with 10% windprofiler input error.

error (which is almost 10 times better than for the unrefined method) and on the accuracy of the vertical z velocity component (more than 10 times better). It can also be observed that 10% windprofiler input error doesn't make the performance much worse, which means that the velocity retrieval with the windprofiler data is less vulnerable to noise.

Now we examine how the retrieved flow changes in the wind profiler region. To focus on the improvement brought by the wind profiler, we use cropped images around the wind profiler radar. Figure 5.4 shows the correct synthetic UV (in the xy plane) velocity field for variation level $K = 5$ and the correct x, y and z component velocity fields as colour images where the colour to magnitude correspondence is given in Figure 1.3b. Figure 5.4 shows the unrefined and refined UV flow for this xy velocity field and U, V and W components separately (side by side for comparison purposes). The refined method uses 10% input error for the wind profiler in this group of data. The black dot in the center indicates the position of the windprofiler radar. The blue circle around this dot is the local area where the windprofiler radar has influence. Compared with the correct flow in the blue circle in Figure 5.4a, the refined flows in the blue circles in Figure 5.4i are qualitatively better than the unrefined flow in Figure 5.4e. The correct velocity flow components, U, V and W, in Figures 5.4b, 5.4c and 5.4d also display their magnitudes in colours. They use the same colour-magnitude correspondence map as in previous discussions but without arrows. The three components of the unrefined corresponding velocity are shown in Figures 5.4f, 5.4g and 5.4h, and the components of the refined one are in Figures 5.4j, 5.4k and 5.4l. We can see big differences in U, V and W from the refined component velocity as compared to the corresponding unrefined ones. Visually, the colour patterns inside the refined blue circles correspond much better to the colour patterns inside the correct blue circles than do the colour inside the unrefined blue circles. We can also see that the regularization method helps refined the influence of windprofiler to the outer blue circle areas, which mean the retrieval improved in a larger range. Here we only show the results for elevation one, but optical flow from the higher elevations give similar results.

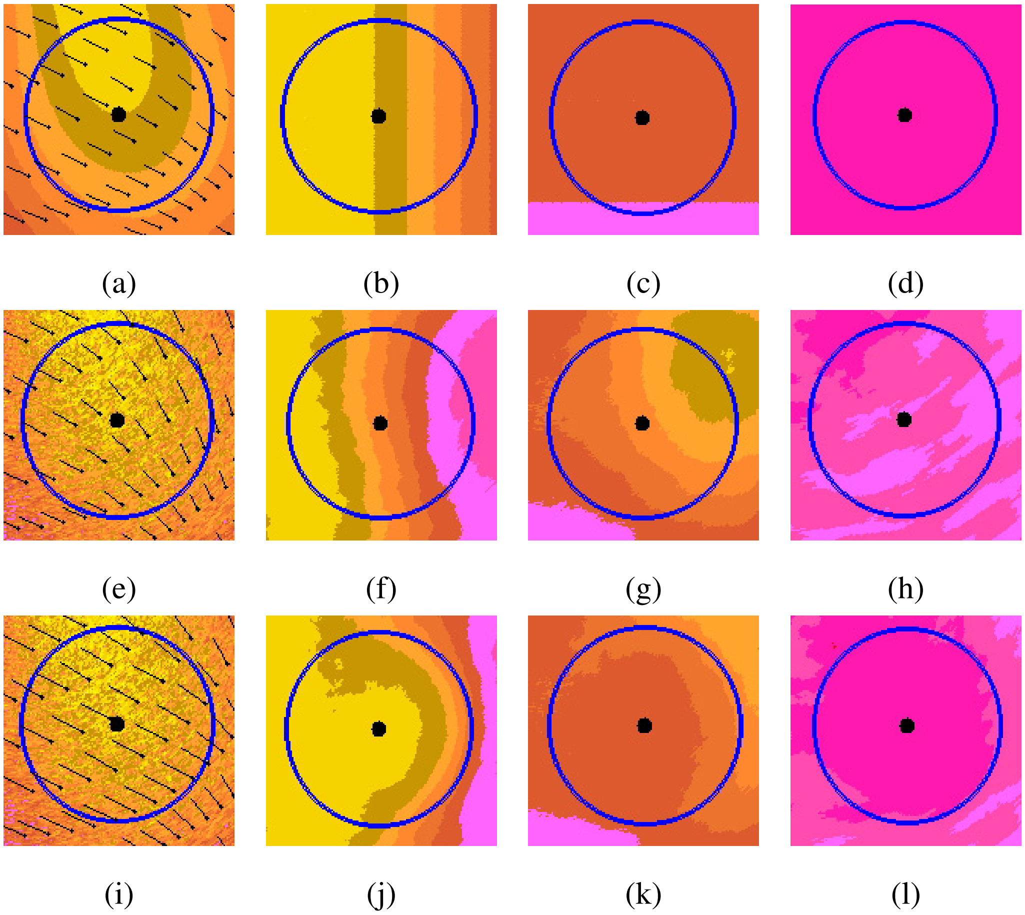

(a) (b) (c) (d)
(e) (f) (g) (h)
(i) (j) (k) (l)

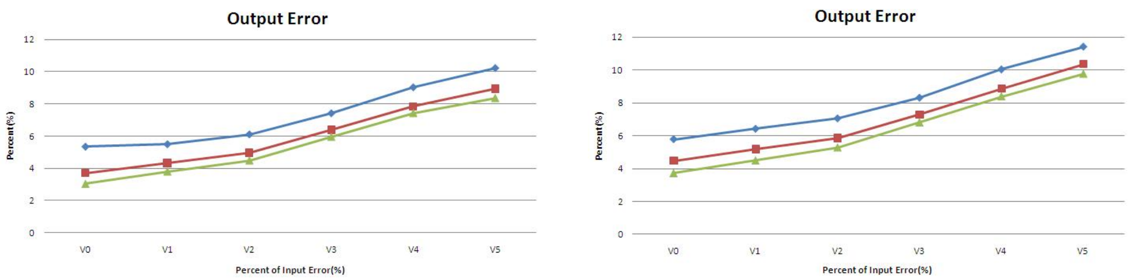
Figure 5.4: The correct synthetic velocity at variation level $K = 5$: (a) the correct UV flow as a vector field on coloured radial velocity, (b) the correct U component, (c) the correct V component and (d) the correct W component. The retrieved synthetic velocity at variation level $K = 5$: (e) the unrefined UV optical flow, (f) the unrefined U component of optical flow along the x axis, (g) the unrefined V optical flow along the y axis, (h) the unrefined W component of optical flow along the z axis, and (i) the refined UV optical flow, (j) the refined U component of optical flow along the x axis, (k) the refined V optical flow along the Y axis, (l) the refined W component of optical flow along the z axis.

5.3.3 The Effects of Parameter Variation on the Synthetic Data

In Equation (5.2) the weight of each windprofiler data point γ_i must be set according to its distance from each Doppler data voxel. Their values are decided by the σ_1, σ_2 and σ_3 as the standard deviations of a 3D Gaussian function. In this section we will investigate the selection of their values and how significant they are in the retrieval performance.

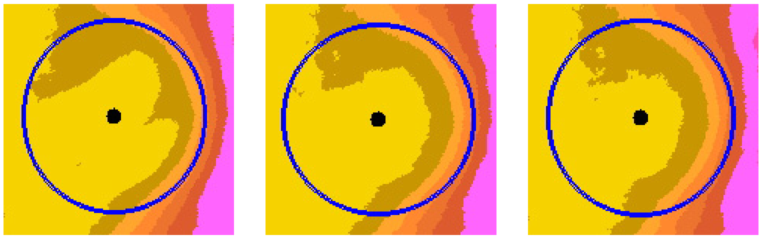
5.3.3.1 The σ Parameter Values

We mainly focus on σ_3, which determines the weight distribution along the z axis on the depth direction, as the selection of the other two can be processed similarly. We examined the performance of the windprofiler radar by setting σ_3 to be 0.4, 0.7 or 1.0, with the six different synthetic velocity variations from $K = 0$ to $K = 5$. Again, the error analysis was only performed for voxels inside the 95% confidence ellipsoidal (inside the blue circle). 20% noise was added to the Doppler data while 10% noise was added to the windprofiler data. Figures 5.5a and 5.5b show the average output errors for different σ_3 values (the blue curve for $\sigma_3 = 0.4$, the red curve for $\sigma_3 = 0.7$ and the green curve for $\sigma_3 = 1.0$) and with Γ in Equation (5.2) set to 1000 in Figure 5.5a and 1800 in Figure 5.5b. It is clear that in each case, $\sigma_3 = 1.0$ gives the best performance. Larger σ_3 values means more data along the z dimension is integrated into the retrieval calculation. The other error metrics (not shown here) show similar results. Figure 5.6 shows the retrieved x component of velocity, U, from the synthetic data for $K = 5$, with the σ_3 values having values of 0.4, 0.7 and 1.0. It can be observed that the size of the "yellow" pattern inside the blue circle decreases as σ_3 increases and that it is a better colour match to the correct U colour pattern shown in Figure 5.4b. This is due to the fact that the σ_3 value determines the shapes of Gaussian distribution in the z dimension. A smaller σ_3 value means only a few windprofiler data points that are close to the Doppler positions have any effects on the computed velocity while a larger value of σ_3 means that many windprofiler data points influence the retrieval near the Doppler positions. Similar conclusions can be shown for the V and W velocity components.



(a) (b)

Figure 5.5: Average output error for (a) $\Gamma = 1000$ and for (b) $\Gamma = 1800$, both with for $\sigma_3 = 0.4$ (diamond curves), 0.7 (square curves) and 1.0 (triangle curves) for the synthetic velocity for the various K values.

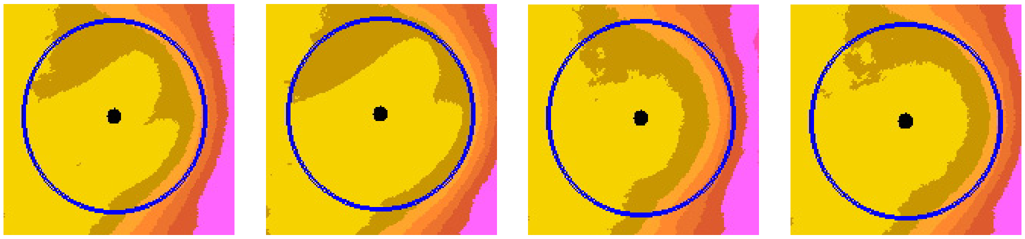


(a) (b) (c)

Figure 5.6: The refined U component along the x axis for synthetic velocity with various σ_3 values for $K = 5$: (a) $\sigma_3 = 0.4$, (b) $\sigma_3 = 0.7$ and (c) $\sigma_3 = 1.0$.

5.3.3.2 The Γ Parameter Values

The value of Γ represents the total weight (influence) of the windprofiler constraint in the retrieval process. Figures 5.5a and 5.5b show that a value of 1800 means the windprofiler radar has almost twice as much influence as a value of 1000. The graphs in Figures 5.5a and 5.5b shows the average output error for $\Gamma = 1000$ and $\Gamma = 1800$ separately. Though the curves look similar, the performance for the $\Gamma = 1000$ case is slightly better than for the $\Gamma = 1800$ case.



(a) (b) (c) (d)

Figure 5.7: The refined U velocity component along the x axis of the synthetic velocity for $K = 5$ with various σ_3, Γ values: (a) $\sigma_3 = 0.4$, $\Gamma = 1000$, (b) $\sigma_3 = 0.4$, $\Gamma = 1800$, (c) $\sigma_3 = 1.0$, $\Gamma = 1000$ and (d) $\sigma_3 = 1.0$, $\Gamma = 1800$.

Figures 5.7a and 5.7b show the x component optical flow (U) for $\Gamma = 1000$ and $\Gamma = 1800$ with $\sigma_3 = 0.4$ at velocity variation level $K = 5$ while Figures 5.7c and 5.7d show the performance for these Γ values for $\sigma_3 = 1.0$. It can be seen that the retrieved colour pattern for velocity magnitudes best matches the correct colour pattern in Figure 5.4c when σ_3 is 1.0 and Γ is 1000.0. This corresponds to the quantitative graph results in Figure 5.5.

The above experiments describe how the performance changes as various parameter values are used. Although our synthetic experiments on the parameter value variations always show that the $\sigma_3 = 1.0$ and $\Gamma = 1000$ parameter values give the best results for our synthetic data, further investigation is necessary before concluding that this will always be the optimal choice. It must be noted that the parameter values we chose may only work well because of the nature of our specific synthetic data.

5.4 Experiments on Radiosonde Data

Besides integrating the windprofiler data as discussed above, our refined regularization approach is also capable of adding new constraints for further refinement. In this section, we show some initial results for integrating radiosonde data that was collected near the Harrow windprofiler radar station in June and July of 2007.

Radiosonde measures various atmospheric parameters from equipment moving vertically in the environment via weather balloons. In the experimental data we used, only the horizontal velocity of wind at different heights is measured, as the balloon rises. No vertical velocity information can be measured. Since it records velocity data at every 1 minute, we treat the data at different heights measured over a short time period as obtained instantaneously and integrate these data into the generalized energy function directly. The new generalized energy function can be expressed as:

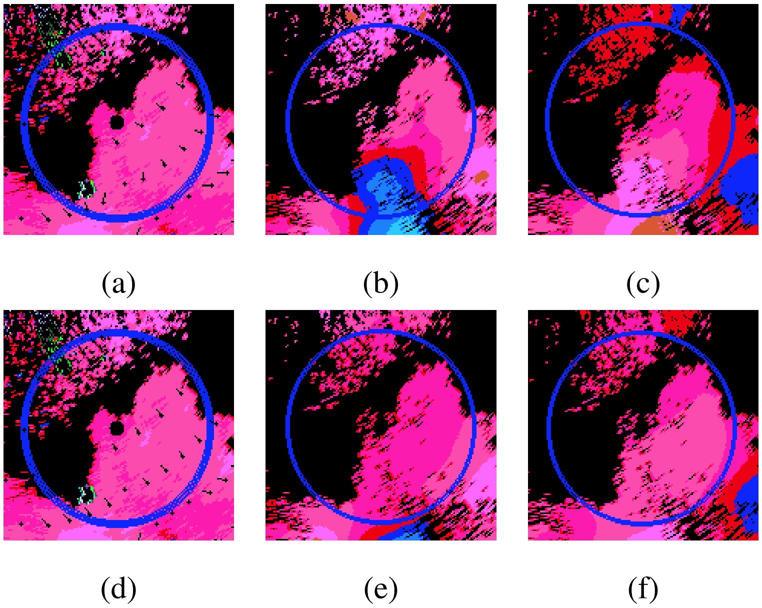
$$\begin{aligned}
&\int\int\int \underbrace{(\vec{V}\cdot\hat{r}-V_r)^2}_{\textit{Radial Velocity Constraint}} + \\
&\alpha^2 \underbrace{(U_x^2+U_y^2+U_z^2+V_x^2+V_y^2+V_z^2+W_x^2+W_y^2+W_z^2)}_{\textit{Velocity Smoothness Constraint}} + \\
&\beta^2 \underbrace{((U-U_{ls})^2+(V-V_{ls})^2+(W-W_{ls})^2)}_{\textit{Least Squares Velocity Consistency Constraint}} + \\
&\sum_{i=1}^{n} \gamma_i^2 \underbrace{((U-U_{wp_i})^2+(V-V_{wp_i})^2+(W-W_{wp_i})^2)}_{\textit{Windprofiler Velocity Consistency Constraint}} + \\
&\sum_{i=1}^{n} \epsilon_i^2 \underbrace{((U-U_{rs_i})^2+(V-V_{rs_i})^2)}_{\textit{Radiosonde Velocity Consistency Constraint}} \;\; \partial x \partial y \partial z,
\end{aligned} \tag{5.3}$$

where ϵ is the Lagrange multiplier for this additional constraint. The value of ϵ_i at each voxel is calculated using the same method based on the 3D Gaussian function as for the other windprofiler constraints:

$$\epsilon_i = \frac{E}{(2\pi)^{\frac{3}{2}} \prod_{k=1}^{3} \sigma_k} e^{-\left(\frac{(x-x_{wp_i})^2}{2\sigma_1^2}+\frac{(y-y_{wp_i})^2}{2\sigma_2^2}+\frac{(z-z_{wp_i})^2}{2\sigma_3^2}\right)}. \tag{5.4}$$

Here $\sigma_1, \sigma_2, \sigma_3$ are the same three standard deviations that specify the shape of the 3D Gaussian distribution according to the distances of radiosonde data point to the calculating Doppler radar data point in three directions, while E is a preset constant value specifying the overall weight assigned on Radiosonde data.

Due to experimental limitations, we don't have data from Doppler radars, windprofiler radars and radiosonde available at the same time. Figures 5.8 and 5.8 show some initial optical flow results from using both Doppler radar and radiosonde data. It can be observed

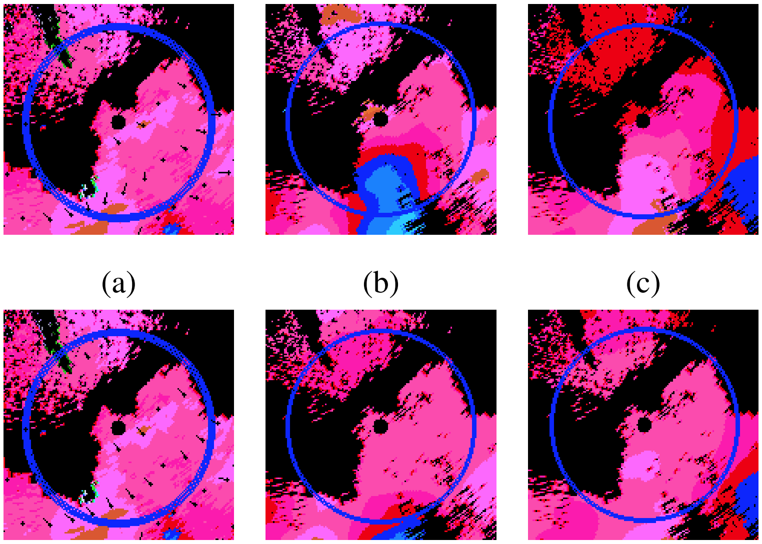

(a) (b) (c)

(d) (e) (f)

Figure 5.8: The horizontal velocity retrieved from Detroit Doppler (and the radiosonde experiment) data on June 28th, 2007 at 00:18:42 for elevation 1: (a) the unrefined UV optical flow, (b) the U component of the unrefined optical flow, (c) the V component of the unrefined optical flow, and (d) the refined UV optical flow, (e) the U component of the refined optical flow and (f) the V component of the refined optical flow.

that inside the blue circle, where radiosonde has influence, the radiosonde data magnitude exceeds the Doppler radar data in both the x and y dimensions. No results on the z dimension are available at present since they are identical. The deviation in the flow in the horizontal XY plane is trivial from the flow obtained from the original solution using only Doppler radar. It shows that our original regularization method provided good results on the horizontal plane.

It is obvious that integrating radiosonde data into the refined solution has very similar effect that resulted from adding windprofiler data. We note that our regularization framework allows further refinement of our retrieved 3D full velocity by introducing other constraints. It should be noted that though the radiosonde data is useful for overall velocity recovery, the main limitation with respect to radiosonde data is that it is only available



(a) (b) (c)

(d) (e) (f)

Figure 5.9: The horizontal velocity retrieved from Detroit Doppler (and the radiosonde experiment) data on June 28^{th}, 2007 at 00:18:42 for elevation 2: (a) the unrefined UV optical flow, (b) the U component of the unrefined optical flow, (c) the V component of the unrefined optical flow, and (d) the refined UV optical flow, (e) the U component of the refined optical flow and (f) the V component of the refined optical flow.

occasionally and the data is limited to the horizontal directions. This drawback limits the application of radiosonde for 3D velocity recovery.

5.5 Conclusions

In this chapter, we "refine" 3D Doppler optical flow by integrating windprofiler data into the calculation. An important result is to make the upwards z components of optical flow near the windprofiler radar much more accurate. We demonstrate this quantitatively on synthetic data and qualitatively on real Doppler data. While the overlap of windprofiler and Doppler is currently only over a small area, we expect that a more dense network of wind profiler radars in the future would provide greater wind velocity accuracy when combined with overlapping Doppler radars. One could imagine the strategic placement of a windprofiler in the vicinity of, say, an airport or city, which, in conjunction with overlapping Doppler data, would lead to more accurate retrieved wind velocity. Similar results can also be found by integrating radiosonde data.

Part II

Storm Detection and Tracking Using Pseudo Storms

Chapter 6

Pseudo Storms

6.1 Introduction

In this chapter we focus on our severe storm tracking algorithm using multiple Doppler radars. First we present an improved **Flood Fill** algorithm to detect storms from the data of several Doppler radars. Second we give a brief introduction to the method to represent storms using **Fuzzy Point Algebra** (interval arithmetic) and then present the original storm tracking algorithm using a **Relaxation Labeling** framework. Due to several drawbacks of the original tracking algorithm when applied to multiple Doppler radar data, we introduce the idea of a “Pseudo Storm” in this chapter and extend the original storm tracking algorithm to a **Pseudo Storm Tracking** algorithm in the next chapter.

6.2 Storm Detection

In our storm detection algorithm. we considered all data points from Doppler radars that have reflectivity values higher than 35 dBZ to be potential storm data. An actual storm is recognized if a cluster of connected storm data points is found. In previous work [17, 41, 56, 68, 69, 70, 82], we used a recursive clustering flood fill algorithm to examine whether the neighbouring storm voxels that have high reflectivity values are connected together. If the size of such connected storm clusters is larger than a threshold, then it is recognized as a storm. This algorithm is recursive, in that it is applied to each unmarked voxel (initially all voxels are unmarked). The algorithm checks whether the reflectivity value is higher than the storm threshold: if it is not, we mark the voxel as “Non-Storm”; if it is then we mark the voxel as “Storm” and apply our flood fill algorithm to all its neighbouring voxels recursively. When all the neighbouring voxels have been recursively marked, if the marked cluster is greater than a preset size threshold we hypothesize the cluster to be a storm. In our earlier 2D work [17, 82, 41] 8-neighbourhood adjacency was used in the recursive step. This was extended to 3D [56, 68] to cover the 26 neighbours adjacent to a voxel (8 in the same plane and 9 in each of the upper and lower planes). However due to the overlapping coverage problem and efficiency issues associated with recursion, the original flood fill algorithm was not used in our current work.

The **Overlapping Coverage Problem**: As shown in Figure 1.5, there are many areas covered by more than one Doppler radar. These radars may have different reflectivity values recorded at or near the same 3D location. If we used the original flood fill algorithm for each of several overlapping radars separately, there could be ambiguous results on whether some areas have storms or not. Recognized storms could also overlap each other. In order to resolve this problem, first we assume a “Storm” marking has higher priority than a “Non-Storm” marking: if a 3D point is labelled as a storm by one radar then it will be labelled as a “Storm” at the same locations in any overlapping radars (regardless of whether these other radars detect a storm voxel there or not). This assumption also holds if radar reports “NULL” data, due to some temporary technical issues while the other overlapping radars work properly. Second, our storm detection algorithm tends to cluster all possible storm voxels together to form a few large well connected storms, as opposed to forming many, poorly connected smaller storms. In order to ensure that there is only one storm marked at each location, we examine all the overlapping data points from all the available radars at the same time. When we mark a data point from one radar, we also check all the neighbouring points around it from the other overlapping radars. Doing this ensures that each data point will be checked only once.

The **Recursion Efficiency Issue**: A recursive algorithm is a programming technique that is easy to understand and simple to implement, but limited by low efficiency in space and time. It requires huge memory space when the recursion depth grows deep (risking an out of memory error). Mover, it is computationally inefficient because the same voxels can be visited many times. The recursive flood fill algorithm is acceptable for one Doppler radar's data since the total volume is restricted. However, this is not the case for the multiple overlapping Doppler radars since the number of radars involved in calculation may be unknown (and is certainly ≥ 2). We replace the original recursive flood fill algorithm with a non-recursive one that uses a stack (a first-in, last-out data structure). In this new algorithm, we still examine each voxel of each Doppler radar. These voxels are pushed into a stack if they are marked as storm voxels. For such voxels, we check all the neighbouring voxels around it in all the overlapping radars, and push any detected storm voxels into the stack as well. As long as the stack is not empty, we pop up the first voxel off the stack and repeat the marking procedure outlined above, until the stack is empty. At this point we have detected a potential storm as a cluster of storm voxels, which we can label with a number. Then, we move to the next unmarked voxel and perform the same marking process all over again. The pseudo code for the algorithm can be found in Algorithm 1 in Appendix C. After all storms have been detected and numbered, we observed that occasionally there were short disconnections in the radar data. We applied a morphological "Dilation and Erosion" procedure to join such storm voxels and get bigger distinct storms. These operations are described below.

6.3 Storm Representation

Krezeski *et al.* [41] first proposed a method to represent the storm as a regular Euclidean point but encountered some problems with this representation. Namely, storm centers did not exactly correspond to Euclidean coordinates, which, in turn, led to jagged storm tracks.

Cheng *et al.* [17] addressed those issues using 2D Fuzzy Point Algebra (2D interval arithmetic). Fuzzy storms are defined as a circle with Euclidean center coordinates. Its intensity is that of its Euclidean center. The radius of the circle was proportional to the storm's size. Fuzzy points allowed some certainty in a point's location because now any point within the circle matches this fuzzy point. Now, Euclidean disparities can be described as fuzzy disparities, using fuzzy magnitudes (an interval between the minimum and maximum magnitudes) and fuzzy angles (an interval between the minimum and maximum angles between temporally adjacent fuzzy disparities). This fuzzy algebra was used in the tracking algorithm.

Qiu [56] extended 2D fuzzy points into 3D by adopting a 3D sphere to represent a fuzzy storm. The radius of the sphere is proportional to the size of storm and the center of the sphere indicates the intensity center of Fuzzy points. In order to represent the real distribution of severe storm points more precisely, Tang *et al.* [68, 69, 70] used a 3D ellipsoid instead of a sphere to describe the uncertainty of a 3D Fuzzy storm's location. Their results demonstrated the efficiency and effectiveness of using the representation to compute storms and their tracks over time. We use the fuzzy ellipsoid storm representation in this thesis.

6.3.1 Calculation of 3D Fuzzy Ellipsoids

We present a brief introduction to how a fuzzy ellipsoid is calculated from storm data. The calculation is based on the assumption that the data of a storm satisfies the distribution of a general multivariate normal density in 3D [26, 68]). If p is one point of this dataset with position at (p_x, p_y, p_z) using cartesian coordinate system, first we can calculate the average

Euclidean center of the storm voxels belonging to the 3D fuzzy storm as:

$$\bar{x} = \frac{\sum_{p \in N} p_x}{\sum |N|}, \tag{6.1}$$

$$\bar{y} = \frac{\sum_{p \in N} p_y}{\sum |N|}, \tag{6.2}$$

$$\bar{z} = \frac{\sum_{p \in N} p_z}{\sum |N|}, \tag{6.3}$$

$$\tag{6.4}$$

where N is the set of storm voxels and $|N|$ is the total number of those voxels. Then we can build the 3×3 covariance matrix of the point density as:

$$\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z^2 \end{bmatrix}. \tag{6.5}$$

The *square root of the variances* (the standard deviations), $\sigma_x, \sigma_y, \sigma_z$ can computed as: calculated as:

$$\sigma_x = \frac{\sum_{p \in N}(p_x - \bar{x})^2}{|N| - 1}, \tag{6.6}$$

$$\sigma_y = \frac{\sum_{p \in N}(p_y - \bar{y})^2}{|N| - 1}, \tag{6.7}$$

$$\sigma_z = \frac{\sum_{p \in N}(p_z - \bar{z})^2}{|N| - 1}, \tag{6.8}$$

and the *the covariances*, σ_{xy}, σ_{xz} and σ_{yz} can be computed as:

$$\sigma_{xy} = \sqrt{\frac{\sum_{p \in N}(p_x - \bar{x})(p_y - \bar{y})}{|N| - 1}}, \tag{6.9}$$

$$\sigma_{xz} = \sqrt{\frac{\sum_{p \in N}(p_x - \bar{x})(p_z - \bar{z})}{|N| - 1}}, \tag{6.10}$$

$$\sigma_{yz} = \sqrt{\frac{\sum_{p \in N}(p_y - \bar{y})(p_z - \bar{z})}{|N| - 1}}. \tag{6.11}$$

We compute the *eigenvectors*, $\hat{e}_i$, and their corresponding *eigenvalues*, λ_i, from the *covariance matrix* in Equation (6.5). We use the three *eigenvectors* as the three axes of the ellipsoid and the square root of the corresponding *eigenvalues*, $\sqrt{\lambda_i}$, as the three radii of ellipsoid.

6.3.2 3D Fuzzy Point Algebra

A set of Euclidean points in 3D space within a 3D ellipsoid can be described as a fuzzy point. This ellipsoid represents an uncertain distribution of a 3D Euclidean points and is called a 3D fuzzy point. According to Tang *et al.* [68, 69, 70], in order to track a storm that is represented as a fuzzy point, we need to use fuzzy algebra. The following definitions related to fuzzy points are required.

A **3D fuzzy point** $E\langle c, r\rangle$ is defined as an ellipsoid with *center* $c = (x, y, z)$, three radii $r = (r_x, r - y, r_z)$ and three *mutually orthogonal* direction vectors $e = (\hat{e}_x, \hat{e}_y, \hat{e}_z)$ as the three axes.

A **fuzzy vector**, $\overrightarrow{E}$, from a fuzzy point E_1 to another fuzzy point E_2 is defined as the infinite set of all displacement vectors from point E_1 to E_2.

A **fuzzy magnitude** of a fuzzy vector $\overrightarrow{E}$ is defined as the set of all magnitudes of all vectors in $\overrightarrow{E}$ and is denoted as $||\overrightarrow{E}||$.

To build a track of storms among a series of images, it is necessary to measure the similarities of storms on sequential images. These measurements include the two disparities of a storm in three adjacent images and the angles between such two disparities. To generalize our discussion, Qiu [56] and Tang [68, 69, 70] all present the definitions of **fuzzy distance** and **fuzzy angle** as below:

A **Fuzzy distance** is defined as the set of all fuzzy magnitudes from one fuzzy ellipsoid to the second one. Usually we present the maximum value and the minimum one as d_{max} and d_{min}.

A **Fuzzy angle** subtended by a non-zero fuzzy vector $\overrightarrow{Q}$ relative to another non-zero vector $\overrightarrow{P}$ is the set of angles subtended by any displacement $\overrightarrow{p}$ in $\overrightarrow{P}$ to another displacement $\overrightarrow{q}$ of fuzzy vector $\overrightarrow{Q}$. A fuzzy angle is denoted as $\langle\overrightarrow{P}, \overrightarrow{Q}\rangle_\theta$.

A fuzzy angle is only calculated when the fuzzy vectors $\overrightarrow{P}$ and $\overrightarrow{Q}$ share a common middle fuzzy point. We need the maximum and minimum values of all angles, denoted as θ_{max} and θ_{min}, to measure the range of angles for a storm that is moving and changing direction. To find θ_{max} and θ_{min} among three ellipsoids requires an appropriate sampling technique applied to all the ellipsoid's interior data points and a searching algorithm with time complexity of $O(n^3)$. This can be very time-consuming and inefficient. In our experimental results, since the storms mainly move from left to right, fuzzy angles don't make a significant difference to our final tracking results but significantly slow down the tracking calculation, therefore we usually disable this measurement. More discussion can be found later in Section 6.4.4.2.

6.4 The Original Relaxation Labeling Algorithm

Similar to the storm detection algorithm above, the tracking algorithm has been developed over two decades. Initially, the tracking of 2D storms as presented by Zhang and Krezeski *et al.* [82, 41] used simple Euclidean points as storm centers. Later, Cheng *et al.* [17, 16] improved this algorithm by using 2D fuzzy storm and fuzzy algebra. Qiu *et al.* [57, 56] then extended this method into 3D using a sphere to represent the storms. Tang *et al.* [68, 69, 70] has modified these spherical fuzzy points into ellipsoidal fuzzy points. The original and our refined tracking algorithms presented in this thesis are based on this ellipsoid fuzzy point definition and its associated algebra.

We define the j_{th} image in the image dataset as I_j with N_j storms in total. The Image I_j's i_{th} storm is denoted as S^i_j. Then we have $\{S^i_j \mid 0 \le i \le N_j \cap S^i_j \in I_j\}$.

6.4.1 The Disparity between Two Storms in Adjacent Images

To track storms over a series of images, first we need to build the connections between two potential matched storms in the consecutive images. This connection is defined as **Disparity** of two storms. It must be noted that not every pair of two storms from the consecutive images will have disparity. We first examine the two storms to find out how similar they are, and can only connect a disparity if these two storms are similar enough. The similarities we checked could be any characters of these two storms, for example whether they have similar sizes, whether in the near locations or how small the rotation angle is between them. The more similar the storms are, the stronger this potential connection between the two storms are. In order to evaluate how strong the connection is, each disparity is assigned with a few values called "certainty" values. Each certainty measure represents the similarity of the two storms in one aspect and is calculated separately. After that, we check each of the certainty values to make sure they are all higher than a preset threshold (we will address this later in more detail in Equation 6.16). If so, then a disparity will be created. then all these certainties will be added together to obtain an "overall certainty" value assigned to this disparity. This overall certainty measure represents the overall similarity between two storms.

Originally, Qiu *et al.* [56, 57] and Tang *et al.* [68, 69, 70] calculated the minimum length, the maximum length, the storm velocity and the storm size, and only used the minimum length and size to check whether a disparity should be built. We have extended their method by adding more certainties to the disparity calculation.

The measurements now used include the storms' size, their positions, the minimum and maximum disparity length, the fuzzy storms' velocities and the Euclidean center coordinate and orientations (of the ellipsoid axes) of the ellipsoids representing the fuzzy storms. Suppose we have two consecutive images I_j and I_{j+1}. For any storm S_j from I_j and S_{j+1} from I_{j+1}, we can calculate the certainty values between any two storms in consecutive images as:

1. The **minimum length certainty** measures how close these two storms S_j and S_{j+1} can be and is denoted as $f_{d_{min}}$.

2. The **maximum length certainty** measures how far the distance between S_j and S_{j+1} can be, and is denoted as $f_{d_{max}}$.

3. The **size certainty** compares the size difference of two storms as:

$$f_s(S_j S_{j+1}) = \begin{cases} 1 - \frac{|r_j r_{j+1}|}{max(r_j, r_{j+1})} & if \quad r_j > 0 \; or \quad r_{j+1} > 0, \\ 1, & \text{otherwise.} \end{cases} \tag{6.12}$$

4. The **position certainty** represents how close the two storms' center positions are:

$$f_p(S_jS_{j+1}) = \begin{cases} 1 - \frac{\|c_j, c_{j+1}\|}{max(r_j, r_{j+1})} & if \ \ r_j > 0 \ or \ \ r_{j+1} > 0, \\ & and \ \| c_j, c_{j+1} \| \leq max(r_j, r_{j+1}) \\ 1, & \text{otherwise.} \end{cases} \tag{6.13}$$

5. The **orientation certainty** is the rotation angle between two fuzzy storms' major radius, denoted as $f_o(\overrightarrow{S_jS_{j+1}})$. First we calculate the acute angle between these two major radii, oS_jS_{j+1}, and then calculate $f_o(\overrightarrow{S_jS_{j+1}})$ as:

$$f_o(\overrightarrow{S_jS_{j+1}}) = 1 - \frac{oS_jS_{j+1}}{\pi}. \tag{6.14}$$

6. The **velocity certainty** measures the similarity of velocities of the storm's center:

$$f_v(S_jS_{j+1}) = \begin{cases} 1 - \frac{\|\overrightarrow{V_j} - \overrightarrow{V_{j+1}}\|}{max(\overrightarrow{V_j}, \overrightarrow{V_{j+1}})} & if \ \ \overrightarrow{V_j} \neq 0,0,0, \ or \ \ \overrightarrow{V_{j+1}} \neq 0,0,0, \\ 1, & \text{otherwise.} \end{cases} \tag{6.15}$$

   where $\overrightarrow{V_j}$ is the full velocity around storm S_j's center and $\overrightarrow{V_{j+1}}$ is the full velocity around storm S_{j+1}'s center.

A disparity will be built if and only if it meets these requirements:

$$\begin{aligned} & f_{d_{min}} = min\|\overrightarrow{S_jS_{j+1}}\| < T_d, \\ & and \ \ f_s(S_jS_{j+1}) > T_s, \\ & and \ \ f_p(S_jS_{j+1}) > T_p, \\ & and \ \ f_o(S_jS_{j+1}) > T_o. \end{aligned} \tag{6.16}$$

where T_d is the minimum distance threshold, T_s is the size threshold, T_p is the position threshold and T_o is the orientation threshold. By choosing appropriate thresholds, we can construct a disparity list L^{j}_{j+1} from Image I_j to I_{j+1} as a set of fuzzy vectors that satisfy all these criteria. Every disparity S_jS_{j+1} in L^{j}_{j+1} is a fuzzy vector between two storms from Image I_j and I_{j+1}. We denote the storm in Image I_j as the disparity's **tail**, and the storm from Image I_{j+1} as its **head**. These disparities all carry their own size certainty, position certainty, orientation certainty with them. Each disparity is assigned an overall certainty value, $f(S_jS_{j+1})$, as:

$$f(S_jS_{j+1}) = \omega_{f_{d\min}} f_{d_{\min}} + \omega_{f_{d\max}} f_{d_{\max}} + \omega_s f_s + \omega_p f_p + \omega_o f_o + \omega_v f_v, \tag{6.17}$$

where $\omega_{f_{d\min}}$, $\omega_{f_{d\max}}$, ω_s, ω_p, ω_o and ω_v are normalized weights that sum to 1. This overall certainty is saved as the initial $p_0(d)$, which will be changed dynamically during the later relaxation labeling procedure, in order to reflect the possibility that this disparity will be selected for the tracks.

6.4.2 Adjacency between Two Disparities

After building the list of all the available disparities, another algorithm is used to connect these disparities to form tracks. A list of "adjacencies" is based on pairs of two consecutive disparities. A particular adjacency has one disparity from the preceding disparity list and another disparity from the following disparity list, with these two disparities sharing a common middle storm. Suppose we have three images I_j, I_{j+1} and I_{j+2}, with two disparity lists L^{j}_{j+1} and L^{j+1}_{j+2}. Assume $\overrightarrow{S_jS_{j+1}}$ and $\overrightarrow{S_{j+1}S_{j+2}}$ are two disparities from the two lists and they have a common storm S_{j+1} in Image I_{j+1}, then an adjacency, A_{j+1}, will be built using them. We denote the first disparity as the adjacency's preceding disparity, and the second disparity as the succeeding disparity. No criteria need to be satisfied to build adjacencies. We compute some "compatibility" values using the certainty values of the two disparities. These compatibility values measure how "strong" the adjacency is. We extended the work of Tang *et al.* [68, 69, 70] to include additional compatibilities values (such as position compatibility):

1. The **minimum length compatibility** C_{dmin}:

$$C_{dmin}(\overrightarrow{S_jS_{j+1}}) = \begin{cases} 1 - \frac{|f_{d1} - f_{d2}|}{\max(f_{d1}, f_{d2})} & if \;\; f_{d1}, \;\; f_{d2} > 0, \\ 1, & \text{otherwise} \end{cases} \tag{6.18}$$

   where $f_{d1} = \min ||\overrightarrow{S_jS_{j+1}}||$ and $f_{d2} = \min ||\overrightarrow{S_{j+1}S_{j+2}}||$.

2. The **maximum length compatibility** C_{dmax}:

$$C_{dmax}(\overrightarrow{S_jS_{j+1}}) = \begin{cases} 1 - \frac{|f_{d1} - f_{d2}|}{\max(f_{d1}, f_{d2})} & if \;\; f_{d1}, \;\; f_{d2} > 0, \\ 1, & \text{otherwise} \end{cases} \tag{6.19}$$

   where $f_{d1} = \max ||\overrightarrow{S_jS_{j+1}}||$ and $f_{d2} = \max ||\overrightarrow{S_{j+1}S_{j+2}}||$.

3. The **fuzzy angle compatibility** C_θ records the maximum fuzzy angle between two storms. It is denoted as $\max\langle\overrightarrow{S_jS_{j+1}}, \overrightarrow{S_{j+1}S_{j+2}}\rangle_\theta$ and recorded as the across of the dot product of the 2 normalized disparities. This is new for adjacency, disparity doesn't have this certainty. It can be calculated as:

$$C_\theta(\overrightarrow{S_jS_{j+1}}) = \begin{cases} 1 - \frac{1+\cos(\max\langle\overrightarrow{S_jS_{j+1}}, \overrightarrow{S_{j+1}S_{j+2}}\rangle_\theta)}{2} & if \;\; d_1, \;\; d_2 > 0, \\ 1, & \text{otherwise} \end{cases} \tag{6.20}$$

   where $\max\langle\overrightarrow{S_jS_{j+1}}, \overrightarrow{S_{j+1}S_{j+2}}\rangle_\theta$ is the fuzzy angle defined between two fuzzy vectors, $\overrightarrow{S_jS_{j+1}}$ and $\overrightarrow{S_{j+1}S_{j+2}}$ as in the previous section.

4. The **size compatibility** C_s:

$$C_s(\overrightarrow{S_jS_{j+1}}) = \frac{f_s(S_jS_{j+1}) + f_s(S_{j+1}S_{j+2})}{2}. \tag{6.21}$$

5. The **position compatibility** C_p:

$$C_p(\overrightarrow{S_jS_{j+1}}) = \frac{f_p(S_jS_{j+1}) + f_p(S_{j+1}S_{j+2})}{2}. \tag{6.22}$$

6. The **orientation compatibility** C_o:

$$C_o(\overrightarrow{S_jS_{j+1}}) = \frac{f_o(\overrightarrow{S_jS_{j+1}}) + f_o(\overrightarrow{S_{j+1}S_{j+2}})}{2}, \tag{6.23}$$

   where $f_o(\overrightarrow{S_jS_{j+1}})$ is the rotation angle between two fuzzy storm's major radius.

7. The **velocity compatibility** C_v:

$$C_v(\overrightarrow{S_jS_{j+1}}) = \frac{f_v(\overrightarrow{S_jS_{j+1}}) + f_v(\overrightarrow{S_{j+1}S_{j+2}})}{2}. \tag{6.24}$$

All the compatibilities above together are called **partial compatibility**. This partial compatibility is calculated as:

$$C_p = \omega_{d\min}C_{d\min} + \omega_{d\max}C_{d\max} + \omega_\theta C_\theta + \omega_s C_s + \omega_p C_p + \omega_o C_o + \omega_v C_v, \tag{6.25}$$

where $\omega_{d\min}$, $\omega_{d\max}$, ω_θ, ω_s, ω_p, ω_o and ω_v are the same normalized weights as in the last section. The partial compatibility and the dynamically updated overall certainties from the adjacency's two disparities are used to calculate the final **overall compatibility** in the later relaxation labeling iteration algorithm. This value will be used to determine whether the disparities are selected into the final tracks.

6.4.3 Relaxation Labeling Algorithm

Once all the candidate disparities are connected by an adjacency, a relaxation labeling process is adopted to refine all the disparities' overall certainty values iteratively to determine the possibility of each disparity surviving in a strong track. In each iteration, the certainty of a disparity will be either updated to be higher if the disparity is more likely to be chosen or updated to be lower if the disparity is less likely chosen compared to the other disparities.

There are two constraints applied when updating a disparity's certainty: The **temporal consistency constraint** and the **special consistency constraint**. The first constraint is concerned with whether a disparity's connections to the previous and following disparities through adjacency are strong. The latter constraint is concerned with whether a disparity's certainty is higher than the other disparities between the same two images. Two variables, specifying **supporting** and **contradictory** evidence, E_s and E_c, with the corresponding two count variables, n_s and n_c, are calculated when applying these two constraints and are later used to update the disparity's overall certainty value.

1. For the temporal consistency constraint case, first we calculate the overall compatibility of each adjacency. Suppose in the k_{th} iteration, the disparity we are examining is d, and there is another disparity, d_t, adjacent to it through a common storm, then the overall compatibility of this adjacency can be calculated as:

   $$C_k(d, d_t) = \omega_1 \cdot C_p(d, d_t) + \omega_2 \cdot (\frac{p_{k-1}(d) + p_{k-1}(d_t)}{2}) \tag{6.26}$$

   where C_p is the partial compatibility saved for the adjacency and p_{k-1} is the current certainty of the disparity. ω_1 and ω_2 are two normalized weights. Then we calculate the average of all the compatibility values from the adjacencies in the same image as T_{ave}. If the calculated $C_k(d, d_t) > T_{ave}$, then $p_{k-1}(d_t)$ is added to the supporting evidence, E_S, and n_s is also incremented by 1. Otherwise, it is added to the contradictory evidence, E_c, and n_c is incremented by 1.

2. For the spatial consistency constraint case, we examine the disparity list which d belongs to and compare its certainty with the other disparities, d_t. If the certainty is higher than d_t, we add the other disparity's current certainty, $p_{k-1}(d_t)$ to E_s and

increment n_s by 1. Otherwise, we add the other disparity's current certainty, $p_{k-1}(d_t)$ to E_c and increment n_c by 1.

3. After we have examined the temporal and spatial consistency constraints, we update the certainty of this disparity, $p_k(d)$ as:

$$p_k(d) = \begin{cases} \frac{1}{2}(1 + \frac{\omega_s E_s - \omega_c E_c}{\omega_s E_s + \omega_c E_c}) & if \;\; E_s \neq 0 \;\; E_c \neq 0, \\ 0, & \text{otherwise,} \end{cases} \tag{6.27}$$

where ω_s and ω_c are weights calculated from the counting variables, n_s and n_c, as $\omega_s = \frac{n_s}{n_s+n_c}$ and $\omega_c = \frac{n_c}{n_s+n_c}$.

As long as the iterations continue, the overall certainty values of the disparities are updated according to the adjacency compatibilities and previous certainties. This procedure ends when the refinement of overall certainty converges to an acceptable level (in our experiment it is that the change of certainty value for each disparity is under 0.0001) or a maximum iteration number has been reached. When convergence has been obtained, we have a set of disparities among all the image storms with stable certainty values.

To build the tracks, for every image, we determine the disparity with the highest certainty and continue this search to its following disparity in next image. This process continues until there are no disparities connected to it through adjacencies. At the same time, when one disparity is selected, all the other disparities having the same storm as its head or tail are deleted. This is because one storm can only be selected by one pair of disparities, one using it as a head endpoint and another using it as a tail endpoint. This procedure is iterated until no other disparity is available in the current image. In this way, all the storms are connected to the storms in neighbouring images that are most likely to be matched. A few tracks of the storms through all the images are formed as well. This procedure is incremental: the relaxation will be repeat whenever a new image I_{N+1} is added to the set of images. In this case, convergence is usually achieved after a few iterations. In the original tracking algorithm, the tracks are displayed by cubic β-splines that go through the storms in each track [56].

6.4.4 Original Results and Limitations

We present the tracking results using the original relaxation labeling algorithm on multiple radars. The data are obtained from two Doppler radars at Detroit and Cleveland on August 20^{th}, 2007 from 12:38:15 to 15:13:17. The sampling resolution is about every 6 minutes, and we use 27 images collected per radar in the image dataset. The precise locations of the radars can be found in Figure 1.5. Qiu *et al.* [56, 57] and Tang et al/ [68, 69, 70] used NEXRAD *I* radar data for a single radar. The NEXRAD *II* data that we use present a number of different features, due to the local geographical and meteorological conditions near the Great Lakes area. For example, the size of recognized storms is much larger compared to those NEXRAD *I* data results. In our experiment, whenever there is a severe storm acclaimed, there is always one or more major storms that occupy the most part of our radar data image, while this seldom happens in the NEXRAD *I* data. As a result, often in our experiments, a single radar is not able to to the whole shape of storm. We have made several changes to the original storm display scheme and tracking algorithm according to these features.

6.4.4.1 Modifications for the NEXRAD *II* Data

First, for display purposes, in order to emphasize storm recognition results, we display coloured information for the recognized storms, as shown in Figure D.1. The large areas of grey colour represents the “Not Storm” data points which have a reflectivity value lower than the storm threshold. The other colours represent the various storms detected, with each storm having its own colour. The colour assigned to each storm depends solely on the order it has been recognized in this image. Since the recognition of storms in each image is independent, no information such as what storms have been detected in the other image is available during the storm recognition procedure in the current image. Thus the storms among images that have been recognized as being the same one in the tracking process may use totally different colours. The black solid ellipsoids drawn on each storm are the fuzzy ellipsoids calculated using the recognized storm parameters. The red points at the ellipsoid center represent the storm's current center with the storm's series number written beside it. The solid blue curves display the tracks retrieved by the original relaxation labeling algorithm and the arrows near one end indicate the final destinations of the tracks with the track's series number displayed beside it.

Second, another change we have made to the display scheme is displaying multiple elevation storm in one image. In the original work by Qiu and Tang, only one elevation's data was displayed per figure. To display the elevations of a radar dataset therefore would require multiple figures. The NEXRAD *II* data has only half the number of elevations compared to the NEXRAD *I* data. Furthermore, it is very common to only use the first few elevations near the ground for the real data. Elevations near the ground usually contain the best data. Higher elevations (for example, 4_{th} and 5_{th} elevations) contain very limited useful data about a storm. In this thesis, we therefore mainly focus on the movements of storms in the horizontal planes that are nearly parallel to the ground level. We choose a "Projection View" method to display the data, which merges the storm data in the vertical direction into one plane. Now, rather than displaying storm data per elevation in the vertical direction, we display the storm data as a whole in one figure, by projecting the data from all the elevations onto one horizontal plane at the ground level. The "Storm" points in any elevation may overlap the "Not Storm" ones in the other elevations. Since the recognized storms do not overlap with each other in our data, there should be no contact between recognized storms.

6.4.4.2 The Disabled Velocity and Fuzzy Angle Constraints

Tang *et al.* [69] showed that optical flow velocity was a good predictor for the displacement of storms in adjacent Doppler datasets. They used the velocity closest to the center of the fuzzy ellipsoid representing the storm as the storm velocity (displacement for one time instance). They used 1999 3D Doppler NEXRAD *I* dataset with a large oblong storm moving from top left to middle right in the images. This storm was completely inside all the images. At each image, the fuzzy ellipsoid representing that storm was projected into the next image and the percentage overlap between the projected ellipsoid and the new ellipsoid now representing that storm in the current image was computed. Since a closed form calculation for intersecting storms does not exist, a simple counting scheme using the equations of the 2 ellipsoids was used to compute the percentage of common voxels (ellipsoid overlap). It was found that the ellipsoid overlap ranged from 87% to 95%, suggesting that optical flow was a good approximation to storm displacement. When the storm velocity was doubled and used to project ellipsoids over 2 frames the overlap dropped to about 40%-50%, suggesting optical flow displacements were only useful for adjacent images. Tang et al. [69] introduced the velocity compatibility factor given in Equation (6.24) above into the original storm tracking algorithm as a way of taking optical flow velocity into account.

However, in our datasets, the above conditions do not hold. While our storms are generally moving left to right, storm, data is continually being introduced on the left and leaving on the right. Thus, while the storm velocities are non-zero and represent the storm motion, they can not be used to track the lager storm centers, which now tend to move erratically in all directions.

There are a few well isolated smaller storms in the Great Lakes datasets that exist for short time periods where optical flow displacement would be a good storm motion predictor but we did not test this. We did use the velocity compatibility function in our initial thesis work but we found it to have little positive effect (but no negative effect) on the tracking. This compatibility function only checks if the velocities on a storm track are roughly the same and not if the storm centers are displaced by this amount (which would require storms to be fully contained and well isolated in all images). The optical flow velocities do not correlate with storm center displacement for our Doppler datasets. Hence, we believe our decision not to use the velocity compatibility algorithm in our tracking algorithms makes good sense for our data.

Similarly, since all the storms move left to right, the fuzzy angle compatibility had little or no positive effect (but no negative effect) on the tracking results but was computationally expensive to compute, due to the larger size of storms in our current datasets compared to the datasets used by Qiu and Tang, so not using it also makes good sense for our datasets.

6.4.4.3 Selection of Parameters

The performance of our relaxation labeling algorithm depends on the selection of all the various parameters. These parameters were chosen by trial and error in the previous experiments of Qiu *et al.* [56, 57] and Tang *et al.* [68, 69, 70]. We also test these history thresholds used previously in our experiments. We choose the same reflectivity threshold for a storm as 35 dBZ as Qiu *et al.* and Tang *et al.* did. We also required that any valid storm have a radius greater than or equal to 5 *km*.

In the disparity building process, the distance threshold, T_d, is set to be 20.0 *km* based on the coverage distance of the NEXRAD *II* radar. The size threshold, T_s, is set to 0.60 of the maximum radius of two storms, so a disparity can only be built when the sizes of two storms are really comparable. The position threshold, T_p, is set to 0.60 to ensure that the distance between two storms is small. The orientation threshold, T_o, is set to 0.40, since the orientation compatibility is less important than other compatibility functions.

The normalized weights, $\omega_{f_{d\min}}$, $\omega_{f_{d\max}}$, ω_θ, ω_s, ω_p, ω_o and ω_v used in disparity certainty and adjacency partial compatibility calculations are also selected through iterated trial and error. The minimum and maximum weights are set to 0.1, the position and orientation weights are 0.2, and the size weight is set to 0.4. As we discussed before, the fuzzy angle and velocity criteria have been disabled in our experiment, therefore ω_θ and ω_v are set to 0.0. In the iterative relaxation labeling algorithm, the normalized weights ω_1 and ω_2 that are used to calculate the overall compatibility value of an adjacency are set to 0.6 for the partial compatibility part and 0.4 for the updated disparity certainty part. The convergence threshold, to decide whether the iterations have converged, is set to 0.0001.

6.4.4.4 Original Tracking Results

Figure D.1 in Appendix D shows the original tracking results for our data. We can see how the storms' centers are moving along each track and how those tracks change over time. There are three major storms (the storms marked as blue, light green and dark green) as well as a few other smaller ones in Figure D.1aa. The track for the blue storm tends to move to the right; the track for the dark green storm tends to stay where it is; the track for the light green storm's changes dramatically, as we see in following images. In Figure D.1ab the previous dark green storm has split into two parts: a red storm and another smaller dark green one. The smaller dark green storm follows the same track as the previous dark green storm while the red one fails in getting tracked since it only lasts for two images (Figures D.1ab and D.1ac). In the later images, we see some very complicated changes in the area where the current dark green storm remains. The storms around this local area go through a lot of merging and splitting before gradually disappearing in Figure D.1al.

Meanwhile, the red and light green storms in the first figure also go through a very complicated development process: first, they follow their own tracks for a few images and then are merged together in Figure D.1af. This merging causes the termination of the track for one storm and that another track has to "jump" back to the new bigger storm's center to follow this change. This merging only lasts for two images (Figures D.1af and D.1ag) and then the merged storm splits into two smaller ones again. The splitting also doesn't last very long, only in two images (Figures D.1ah and D.1ai). We cannot build a track for these smaller storms since a valid storm must last longer than two images for an adjacency to be built. The major track of the bigger storm again has to "jump" to a new position as shown by the red point, which indicates the new storm's center. This merging and splitting process continues until in Figure D.1am, where the storms finally merge together into one huge storm. This huge storm is very stable and almost lasts until the end of the dataset. From Figure D.1am to Figure D.1ay, this huge storm continuously moves to the right. Its coverage area gradually decreases until in the last two Figures D.1az and D.1ba the storm splits into two parts with the right part forming a new storm and the left part remains recognized as in the track of the old huge storm.

6.4.4.5 Limitations of the Original Tracking Algorithm

From the above discussion we see that there are some serious problems in the original algorithm's tracking results. First, from all the figures, we can see that the clouds are moving to the right in the long term. The tracks we retrieved using the original algorithm generally reflect the real storms' trajectory as long as the storms that the tracking is based on are relatively "independent". By independent we mean that the storms do not interfere with each other and that no merging or splitting storm events occurs. However, the correct tracks do not always satisfy these constraints. The consistency of a storm trajectory is always disrupted by random changes of clouds' shape, caused, for example, by the sudden merging or splitting of the storms. Then the track has to "jump" to another position to suit the abrupt changes. This is all because the real clouds move relatively slowly and tend to remain where they are. However, their shapes change significantly over time (this is why Euclidean point tracking doesn't work). Our storm detection method, the **Flood Fill** algorithm, works solely depending on the connectivity of storm points in the real data and therefore could determine a totally different number of storms when these random changes occur. Thus, the original tracking algorithm that calculates its storm trajectories of storms detected by the flood fill algorithm is negatively influenced by this uncertainty in real weather data.

Therefore, we designed a new algorithm that can overcome the original algorithm's drawbacks in this thesis. We developed an advanced algorithm that can accommodate the uncertainty in real weather storms' shapes and positions. The algorithm does not depend just on the connectivity of storm data points but also on the history of storms and future predictions based on this history information. Our new algorithm is able to accommodate for random changes in a storm's shape, size and position to generate better tracking results. Inspired by the work of Krezeski *et al.* [41] the idea of "Pseudo Storm" is re-introduced and significantly improved.

6.5 Pseudo Storms

The concept of a “Pseudo Storm” is used to remove/attenuate the limitations of the original tracking algorithm as described above. Our objective is to remove abrupt changes in storm trajectories, accommodate merging and splitting of storms and generate smooth trajectories that reflect a real storm’s motion. The original storm tracking algorithm’s problems are caused by the rapidly changing morphological properties of storms. Our solution to this problem is adding a significantly modified notion of “pseudo storms” (introduced by Krezeski et al, [41] to our tracking algorithm. A pseudo storm can be considered to be a collection of one or more real storms that need to provide a smooth track and account for the data. By adding pseudo storms to the set of potential storms to be tracked, we provide alternative choices for the tracking algorithm to select. Now, if the initial real storms don’t produce good trajectories, we can use pseudo storms in place of the comprising real storms instead. As a result, we are more likely to obtain good trajectories than before.

6.5.1 Designing Pseudo Storms

First, we present the definition of pseudo storm and then introduce modifications to the original tracking algorithm to recognize them.

The concept of a Pseudo Storm is defined as:

Definition If in a current image, two or more real storms have roughly the same coverage as another bigger real storm in the previous or following image, then we group these real storms in that current image together and consider them as a **Pseudo Storm**.

We consider information from the previous image and following image as equally important in storm trajectory calculation. We limit the number of images we consider when generating pseudo storms to be the direct neighbouring images just before or after the current image. We only consider adjacent images to provide a fast reaction to the ongoing development of severe weather storms.

6.5.2 Implementation of Pseudo Storms

We describe the desired behavior of a pseudo storm based on our observations of the real storms and expected tracking results as:

1. From our observation of real data, we note that the real velocity of a storm is quite slow. During the time interval that we have used for our experiments (about 5 ~ 6 minutes), storms in the next adjacent image always overlapped with the current storm. Therefore, when hypothesizing a pseudo storm we need only check all the pseudo storm's real storm centers to see if they are in another real storm's coverage in the neighbouring images.

2. It is only one prerequisite in the generation of a pseudo storm that the cluster of real storms' centers share another real storm's coverage in an neighbouring image. We also check whether the sizes of storms matches. First, we examine the overall coverage (volume), V_p, of all sets of storms to see whether the overall coverage is compatible to a real storm in the neighbouring images, V_n. Second, not all storms in the same coverage area will be included in a pseudo storm. The storms that are too small are ignored because they cannot produce any obvious difference in the storm trajectories by adding them to pseudo storm . In our experiments, we set the ratio threshold of the overall volume of pseudo storm, V_p, to the volume of the referring real stormV_n as $T_{pn} = 0.7$. As well, we required that all candidate storms of a pseudo storm be at least 10% of real storm in the neighbouring images.

3. To simplify our definition of pseudo storms, we disallow the merging/splitting of pseudo storms into other pseudo storms. That is, we assume that pseudo storms can only be generated from **real** storms. Pseudo storms cannot merge into bigger pseudo storms and a pseudo storm cannot split into several smaller pseudo storms in neighbouring images. The generation of a pseudo storm is based solely on real storms.

4. As discussed earlier, we generated a pseudo storm by checking the previous and following images. It is possible that these two images provide conflicting information about which storms should be considered for inclusion in a pseudo storm. We treat the two directions equally important for the generation of a smooth track. To resolve such conflicts, we allow one real storm to belong to two pseudo storms at the same time: one based on the previous image, and the other based on the following image. To distinguish these two pseudo storms, we define them as **pre-pseudo storms** and **next-pseudo storms**. These pre-pseudo storm and next-pseudo storm are allowed to overlap with each other in neighbouring images but in the same image no two pre-pseudo storms or two next-pseudo storms can overlap.

Based on all these rules, we present the algorithm to generate pseudo storm from real storm data. In each image, we first go through all the recognized real storms, comparing their centers' position with the previous image. If two or more storms overlap with the same storm's coverage in the previous image, we record these storm candidates as potential pre-pseudo storms. We make the same comparison with the next image, again generating potential next-pseudo storms. Given this set of pseudo storms, we examine their eligibility as pseudo storms by seeing if their sizes are within 10% of the overlapping storms in previous and following images and if the absolute value of their overall volume differences is larger than or equal to T_{pn}. If these conditions are satisfied, then we have hypothesized a pseudo storm. The calculation of a pseudo storm's center and three radii uses the same fuzzy point algebra as for real storms. The detailed algorithm can be found in Algorithm 2 in Appendix C.

6.5.3 Pseudo Storm Tracking Results using the Original Tracking Algorithm

In this section, we present the trajectories found by adding pseudo storms to the real storm data using the original tracking algorithm. This required some modifications to the existing algorithm. For example, if a pseudo storm is in a potential storm trajectory, then the disparities using the real storms inside this pseudo storm between the same images must be disabled. Similarly, if a disparity connecting two real storms is selected, the disparities having conflicts with rejected pseudo storms must also be disabled. In this way we guarantee that any storm point can only belong to one track at the same time. The results are displayed in Figures D.2aa to D.2ba in the second part of Appendix D for the Great Lakes Doppler radar reflectivity data.

In the first two Figures D.2aa and Figures D.2ab there are no pseudo storm detected. In Figure D.2ac one pseudo storm is detected and is marked with a red dashed ellipsoid over the orange and dark green real storms. This pseudo storm is obtained from the real storm marked as dark green in the next image shown in Figure D.2ad. It is interesting that in Figure D.2ad there is also one pseudo storm recognized near the same area, and this pseudo storm is actually obtained based on the real storm marked as dark green in the previous image, as shown in Figure D.2ac. Thus in these two images, pseudo storms based on each other's real storms have been hypothesized. The two real storms in Figure D.2ac have merged together as one real storm, as shown in Figure D.2ad. At the same time, another storm has split from the merged storm, as shown in Figure D.2ad. We can see that these pseudo storms truly reflect the complicated changes of the storms over time.

In Figure D.2ae, the large storm in the bottom area and the smaller storm in the upper left area merge together as one huge pseudo storm, marked as the dashed red ellipsoid in the figure. This pseudo storms's generation was based on the huge real storm in the next image, as shown in Figure D.2af. This huge storm continues until it breaks into two parts in Figure D.2ah. Here, the new added pseudo storm has successfully kept the split real storms together. The three major pseudo storms recognized in Figure D.2ah are all based on the real storms in previous image as we found no relevant storm information in the next image. This is a good example demonstrating how a pseudo storm makes the storm tracking process more stable and helps smooth the computed trajectories. The splitting of the huge storm continues in the next image, as shown in Figure D.2ai, where another pseudo storm is detected, based on the real storm in its next image in Figure D.2aj. It must be noted that the continuous detection of pseudo storms in the same area as just described can only last for two images. If the split were to continue to one more image, there will not be any pseudo storm generated in the middle image since pseudo storms cannot be generated from other pseudo storms. In that case, the storm trajectory would be broken and the sub storms will be tracked separately.

If we look at the upper left part in Figure D.2al, the two small pseudo storms detected there overlap one real storm. This is a good example showing how one real storm can be part of two pseudo storms at the same time. These two pseudo storms are generated according to the previous image and following image separately. The pre-pseudo storm helps to record the previous image's information while the next-pseudo storm helps in building track to the next image.

Looking at the middle bottom part of Figures D.2ak and D.2al once again, we see that the huge storm breaks into two parts, but the track is successfully computed using the pseudo storms in these two images. The real storms that belong to the huge pseudo storm have finally merged together, as one huge real storm, marked in green, in Figure D.2am. This storm remains very stable as one huge real storm moving steadily to the right until the last images are reached. During this storm stability period, as shown in Figures D.2am to D.2ay, there are no pseudo storm detected. By Figure D.2az the huge storm has split into two parts again, so one pseudo storm is generated based on previous image information to keep smoothly tracking the storm. At this time, there is a storm marked in pink that splits from the huge storm's bottom right part. This change continues in the next image, which is also the last image in the sequence and is shown in Figure D.2ba. The pseudo storm combining the huge green real storm and the newly split pink storm in the previous image has accomplished its task and has, therefore, disappeared. If the tracking process were continued (if the sequence was longer) to the next image, it is very possible that a new track would have been created, based on this new split storm.

If we compare the tracks using pseudo storms with the ones from original method using only real storms, it is very clear that there are more tracks found and that those tracks using pseudo storms are shorter. The largest track in the original real storm trajectory has been split into several parts now. As shown in Figure D.2ae, the biggest real storm, marked as green in the bottom of the image, has its own track marked as a very short split since it doesn't move very much. The pseudo storm that is created based on the next image also generates its own track starting from this image. This new track is very long, indicating how the huge storm will develop over a long time period. However, these two tracks cannot be linked together since they don't share any common storm. The pseudo storms we generated here match the real storms in the neighbouring images much better than the real storm counterpart, which demonstrates that pseudo storm storm tracing helps smooth the trajectory. However, it also has the disadvantage of breaking the original tracks into several parts, whenever a pseudo storm is adopted. To connect the real storms and the pseudo storm, we need to modify the original data structures that we used in the tracking algorithm, such as the current disparity and adjacency models that right now can only represent a one-to-one relationship. The original relaxation labeling algorithm has to be redesigned (as discussed in the next chapter) to deal with more complicated scenarios such as this.

Chapter 7

Pseudo Storm Tracking

In this chapter we introduce a new tracking algorithm for tracking pseudo storms and demonstrate its performance on the Great Lakes Doppler radar datasets.

7.1 Pseudo Storm Tracking Algorithm

In the preceding chapter, we demonstrated the principle of the original tracking algorithm without and with pseudo storms. The pseudo storms there do help improve the smoothness of storm tracking, by using a pseudo storm instead of a group of real storms. However as we discussed previously, it also cuts the tracks into pieces, since the tracks generated there are limited to only connecting one disparity to another one. Therefore the disparities from a pseudo storm and the disparities from its comprising real storms can not be connected in the original method. Here we present an advanced storm tracking algorithm taking full advantage of the newly generated pseudo storms, which not only build the traditional $1-1$ adjacency but also new $1-n$ relationship between storms.

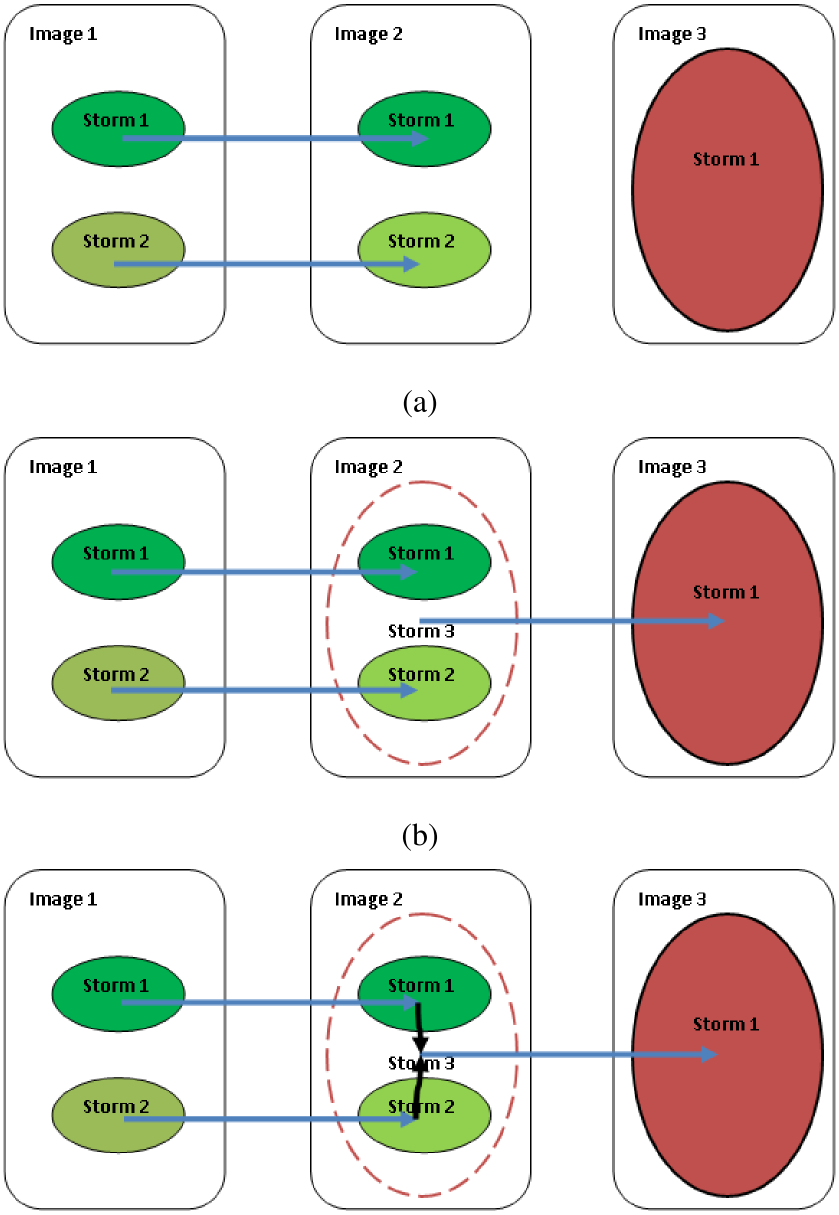
In the original storm tracking algorithm, in order to generate tracks with pseudo storms, we first built the disparities between storms and then used these disparities to generate the adjacencies. The certainties of disparities to stay on a chosen track are updated using an iterative relaxation labelling program. After this calculation, we build the storm trajectories from all disparity certainties and save them for later display. In this chapter, we first investigate all the changes brought by the pseudo storms and determined the necessary modifications required to apply pseudo storms. Secondly, we discuss how to define the new structures of disparity, connectivity and adjacency. Then we present the new relaxation tracking algorithm. We also address how to create tracks from chosen disparities and how to record all the tracks efficiently.

7.1.1 Algorithm Design

The calculation of a "disparity between two storms" remains the same as in the original storm tracking algorithm. The differences between two adjacent storms is compared via several metrics such as minimum and maximum fuzzy length, fuzzy size, fuzzy orientation, fuzzy position and and fuzzy velocity. However, we added a new data structure to the adjacency model to record the "connections" among adjacencies. This connection data structure records the links between pseudo storms and the real storms that comprise them.

7.1.1.1 Building Connectivity

We examine a simple example shown in Figure 7.1a, where the data consist of three images in a short sequence: Image 1 and Image 2 both have two real storms and Images 3 has only one bigger storm that the two storms in Images 1 and 2 merge into. Figure 7.1a shows the disparities using the original storm tracking algorithm. There are only two disparities that can be recovered from Image 1 to Image 2. The size changes of the storms from Image 2 to the storm of Image 3 are too big to allow a disparity to be built using them. So there is no adjacency built among these three images, since adjacency requires at least two disparities to be connected consecutively. In Figure 7.1b we add a pseudo storm in Image 2 based on the single larger storm in Image 3, marked by the dashed red ellipsoid. There is a disparity created between the pseudo storm in Image 2 and the real storm in Image 3. However, there is no common storm between the new disparity and any of the two older disparities, so we still cannot build an adjacency based on them. To solve this problem, as shown in Figure



(a)

(b)

(c)

Figure 7.1: (a) Artificial tracking result using original storm tracking tracking algorithm; (b) Artificial tracking result using pseudo storms and the original storm tracking algorithm; (c) Artificial tracking result using pseudo storm and a connection between a pseudo storm and a real storm.

7.1c, we add a new relationship to connect the disparity from the pseudo storm and the disparities from its existing real storms. We define this new relationship as a "Connection" between a real storms and the pseudo storm they belong to. The connection is drawn as a black arrow in Figure 7.1c. The detailed definition is given as:

Definition Given three consecutive images, we build a **Connection** between two storms in the middle image if one storm is a real storm and the other storm is pseudo storm that contains this real storm. The connection is only built if there is at least one disparity connected to the pseudo storm and at least one other disparity connected to the real storm and these two disparities are between different sets of images.

Our connectivity definition doesn't care if the disparities that are connected are head or tail storms of the disparities. But the disparities have to be in the same direction in time and they must distribute in different images. For example, assume there is a pseudo storm A and a real storm B that comprises it in the middle image. If the disparity connected to pseudo storm A is in the direction to A, then the disparity connected to the real storm B has to be moving away from this real storm to another storm in the next image; similarly, if the disparity connected to A is using it as tail storm that is pointing away from this pseudo storm, then the disparity to the real storm B must move away from a real storm in the previous image. After building the connections between pseudo and real storms, the disparities here will be treated equally as the other disparities that share one common storm as in the original tracking algorithm. Then we can build an adjacency relationship between these disparities.

It should be noted that the reason why we build the connection between a pseudo storm and the real storms to link disparities, rather than connect these disparities directly is because of the processing efficiency issue. In the case where there is more than one disparity connected to the same storm, we only need to build one connection between the storms and let all the disparities share this connection together, which is simpler than building connections for each pair of disparities.

We only build a connection between a pseudo storm and its corresponding real storm when they both have available disparities to create an adjacency. Otherwise, the connection we build will be useless in the tracking algorithm. Furthermore, Figure 7.2 shows a more complicated case than in Figure 7.1, where the pseudo storms overlap each other on one of the real storms that comprise them. Figure 7.2a shows the disparities built up using the original tracking algorithm. Figure 7.2b shows the pseudo storms we detected based on real storm information, in which we can see two pseudo storms having an overlapped real storm in its middle image. One pseudo storm is detected from the previous image and the other one is based on the next image. Figure 7.2c shows how our new algorithm creates connections with these two pseudo storms, where the disparities between real storms and pseudo storms now are connected by the connections marked as black arrows. It can also be seen that the storm 1 in the middle image, Image 2, has no disparity connected to it, and so no connection has been built on it.

7.1.1.2 Building Adjacencies using Pseudo Storms

The adjacencies in our tracking algorithm can be divided into two classes. The first class of adjacency is the same as in the original storm tracking algorithm for connecting two adjacent disparities. It doesn't matter what kind of storms the disparity connects: the storms can both be real, one storm could be real and the other a pseudo storm or the storms are both pseudo storms. As long as the two disparities have a common storm in the middle image, then no connection is necessary to create this adjacency. The second class of adjacency is created whenever two disparities are connected through a built connection as in the previous section. On one side of a middle image there is one (and only one) disparity connected to the pseudo storm. On the other side of the middle image there is at least one disparity connecting the real storms that comprise the pseudo storm. A typical data structure of this new adjacency is described as in Figure 7.3a.

It is clear that in the second class adjacency there is only one disparity connected to the pseudo storm in the middle image but there could be more than one disparity connected to

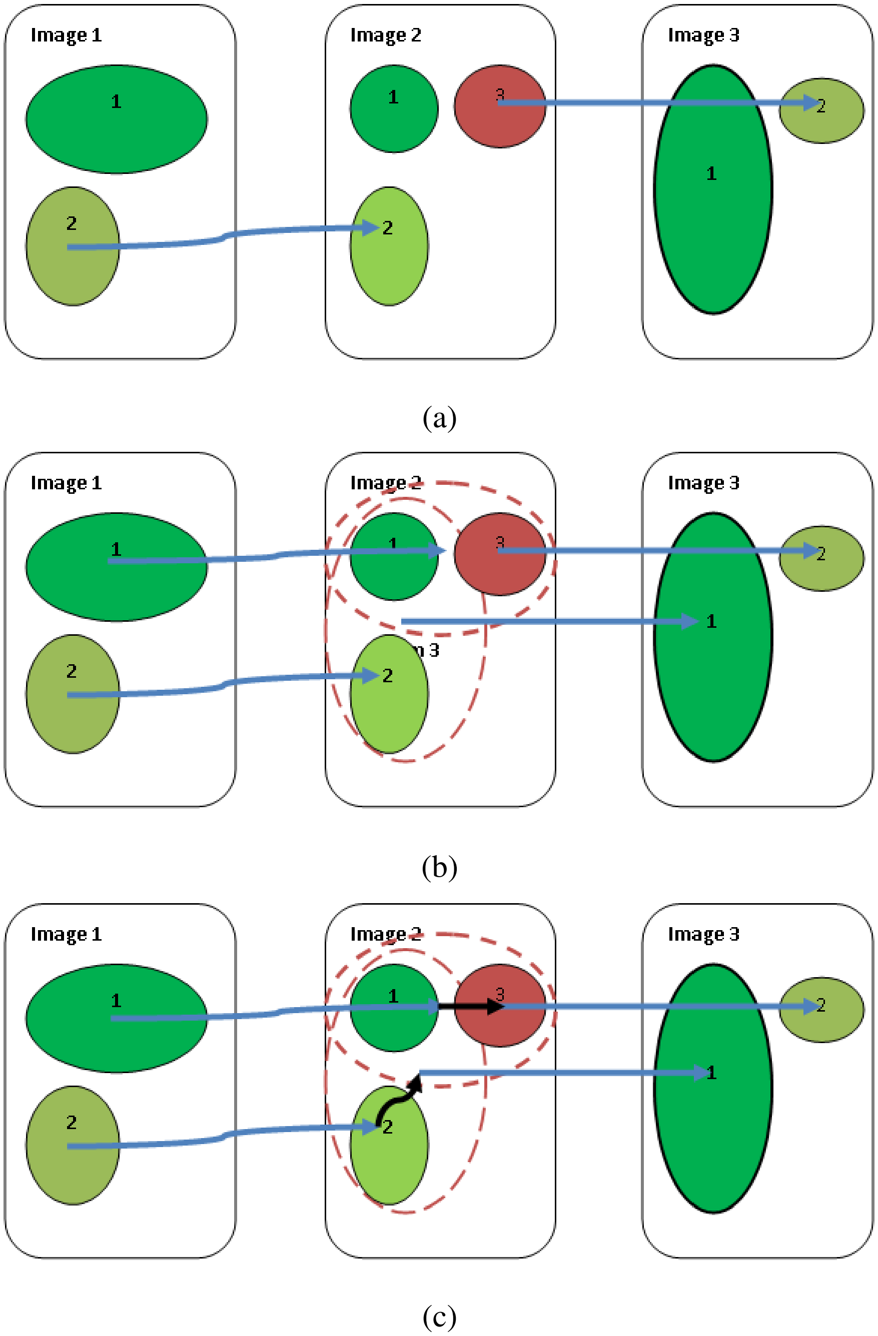
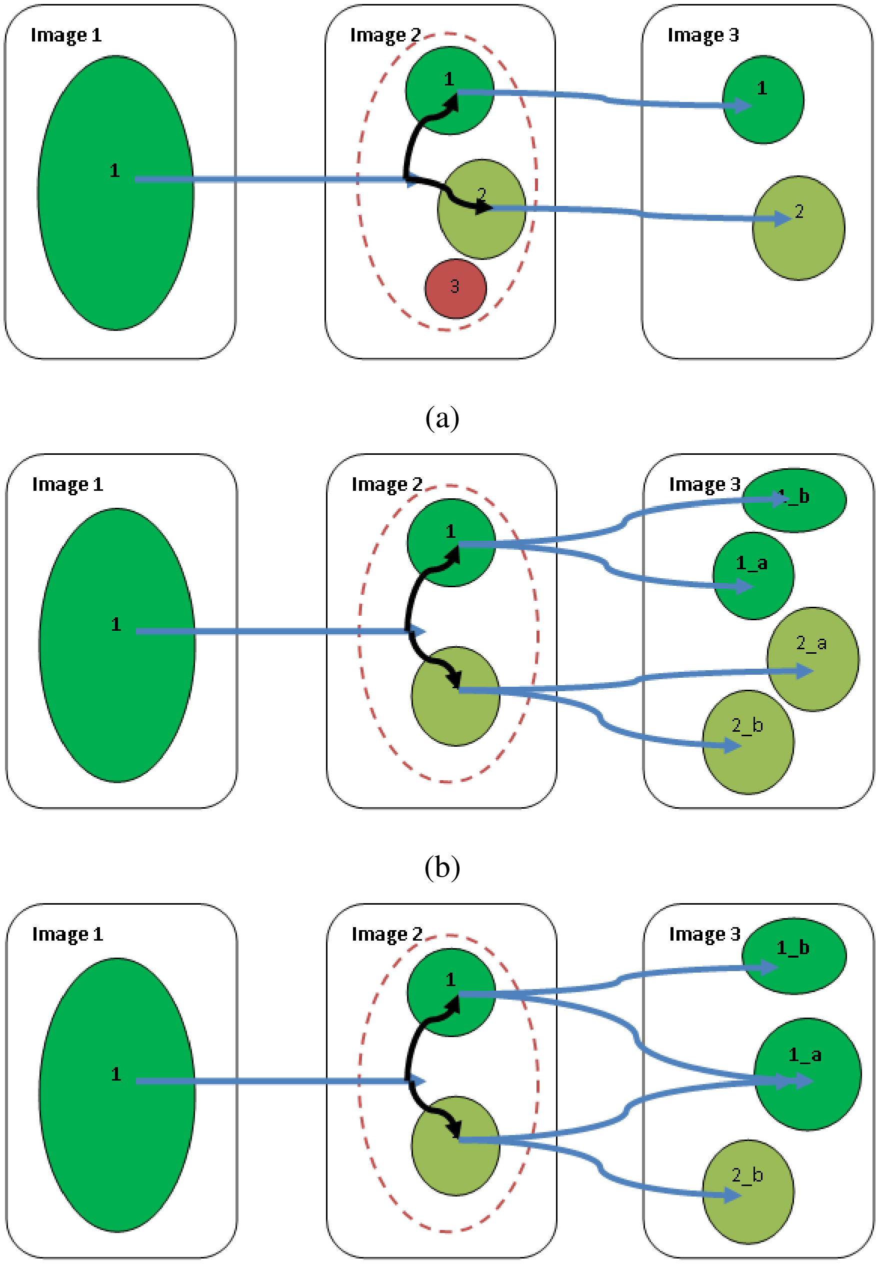


Figure 7.2: Artificial tracking results for a more complicated case: (a) using the original tracking algorithm; (b) using the original tracking algorithm with pseudo storms; (c) using pseudo storm and our new pseudo storm tracking algorithm.



(a)

(b)

(c)

Figure 7.3: (a) The organization of a typical adjacency with pseudo storms; (b) Building an adjacency where the real storms have multiple disparities; (c) Building an adjacency in a more complicated situation.

the real storms that comprise it. As we have discussed before, if a real storm doesn't have any disparity connected to it, no connection will be created for it. To summarize our discussion, there is no connection between two pseudo storms that share one or more common real storms since this relationship could be very weak and won't bring any improvement to our tracking results.

It should be noted that a pseudo storm may have a real storm that has more than one disparity connected to it, as in Figure 7.3b. In this case, we have to build an adjacency for each possible combination. As shown in Figure 7.3b, a pseudo storm has two real storms, Storm 1 and Storm 2, in the middle image, Image 2. Each of these real storms has 2 disparities connected to different storms in Image 3. In this case, the number of adjacencies we have to build is $2 \times 2 = 4$. When we build the adjacencies, we have to go through all the possible combinations of disparities and build one adjacency for each of them. This scenario could be even more complicated as shown in Figure 7.3c, where the disparities connect to the same storm in Image 3. Since these two disparities map to the same storm, they cannot co-exist in the final storm tracks, so no adjacency is created for this combination case. Therefore, when we build the adjacencies, we have to delete the conflicting cases whenever the disparities for the real storms are pointing to the same storm in the other image.

Once the disparities for an adjacency are decided, we begin to build the adjacency. We define A_{j+1} as an adjacency built among three images, I_j, I_{j+1} and I_{j+2}. Then A_{j+1} will have three parts: the set of disparities from I_j to I_{j+1}, Set_{Disp_j}, defined as preceding disparity set, in which the number of all disparities is $num\{Set_{Disp_j}\}$; the set of disparities from I_{j+1} to I_{j+2}, defined as succeeding disparity set, $Set_{Disp_{j+1}}$, in which the number of all disparities is $num\{Set_{Disp_{j+1}}\}$; and a set of connections to connect these disparities, $Set_{Con_{j+1}}$, in which the number of all connections is $num\{Set_{Con_{j+1}}\}$. Either $num\{Set_{Disp_j\}}$ or $num\{Set_{Disp_{j+1}}\}$ has to be 1, while the other number is at least 1 but can be greater. $num\{Set_{Con_{j+1}}\}$ either equals 0 for the first class of adjacency or equals the largest number among $num\{Set_{Disp_j}\}$ and $num\{Set_{Disp_{j+1}}\}$ for the second class of adjacency.

Since the adjacency structure is changed to have a set of disparities, the calculation of all the partial compatibilities have to be modified as well. For every partial compatibility such as the minimum length, maximum length, size, position, orientation or velocity compatibilities, we first calculate the average of all the disparities in one set, and then calculate the partial compatibility value based on the two average values. The calculation function can be expressed as:

1. The **minimum length compatibility** $C_{d_{min}}$

   $$C_{d_{min}}(\overrightarrow{S_jS_{j+1}}) = \begin{cases} 1 - \frac{|\overline{d_1}-\overline{d_2}|}{\max(\overline{d_1},\overline{d_2})} & if\ \ \overline{d_1},\ \ \overline{d_2} > 0, \\ 1, & \text{otherwise} \end{cases} \tag{7.1}$$

   where $\overline{d_1}$ is the mean of the minimum length d_1 from all the disparities in Set_{Disp_j} and $\overline{d_2}$ is the mean of minimum length d_2 from all the disparities in $Set_{Disp_{j+1}}$.

2. The **maximum length compatibility** $C_{d_{max}}$:

   $$C_{d_{max}}(\overrightarrow{S_jS_{j+1}}) = \begin{cases} 1 - \frac{|\overline{d_1}-\overline{d_2}|}{\max(\overline{d_1},\overline{d_2})} & if\ \ \overline{d_1},\ \ \overline{d_2} > 0, \\ 1, & \text{otherwise} \end{cases} \tag{7.2}$$

   where $\overline{d_1}$ is the mean of the maximum length d_1 for all the disparities in Set_{Disp_j} and $\overline{d_2}$ is the mean of the maximum length d_2 from all the disparities in $Set_{Disp_{j+1}}$.

3. The **size compatibility** C_s:

   $$C_s(\overrightarrow{S_jS_{j+1}}) = \frac{\overline{f_s(S_jS_{j+1})} + \overline{f_s(S_{j+1}S_{j+2})}}{2}. \tag{7.3}$$

4. The **position compatibility** C_p:

   $$C_p(\overrightarrow{S_jS_{j+1}}) = \frac{\overline{f_p(\overrightarrow{S_jS_{j+1}})} + \overline{f_p(\overrightarrow{S_{j+1}S_{j+2}})}}{2}. \tag{7.4}$$

5. The **orientation compatibility** C_o:

$$C_o(\overrightarrow{S_jS_{j+1}}) = \frac{\overline{f_o(\overrightarrow{S_jS_{j+1}})} + \overline{f_o(\overrightarrow{S_{j+1}S_{j+2}})}}{2}. \tag{7.5}$$

6. The **velocity compatibility** C_v:

$$C_v(\overrightarrow{S_jS_{j+1}}) = \frac{\overline{f_v(\overrightarrow{S_jS_{j+1}})} + \overline{f_v(\overrightarrow{S_{j+1}S_{j+2}})}}{2}. \tag{7.6}$$

All of the compatibilities above taken together are called the **partial compatibility**. The overall partial compatibility is calculated as:

$$C_p = \omega_{d_{min}}C_{d_{min}} + \omega_{d_{max}}C_{d_{max}} + \omega_s C_s + \omega_p C_p + \omega_o C_o + \omega_v C_v, \tag{7.7}$$

where $\omega_{d_{min}}$, $\omega_{d_{max}}$, ω_s, ω_p, ω_o and ω_v are normalized weights that sum to 1. The partial compatibility and the dynamically updated certainties from the adjacency's two disparities will be used to calculate the final **overall compatibility** in the later relaxation iteration. This value will be then be used to determine whether the disparities are selected for the final storm tracks.

As we discussed in the previous chapter, the velocity compatibility cannot provide obvious improvement but badly slows down the calculation; therefore we have disabled it in our following experiments. It can also be noted that there is no such **fuzzy angle compatibility** here as in the original tracking algorithm. This is due to the fact that we have changed the structure of adjacency to include more than one disparity on one side of the adjacency. As a pseudo storm breaks into several real storms, it doesn't make sense to calculate the fuzzy angle between the pseudo storm and the real storms that comprise it. Therefore in pseudo storm tracking algorithm we don't have fuzzy angle involved by design.

7.1.1.3 Relaxation Labeling Procedure

The new relaxation labeling algorithm works much the same as the original relaxation labeling algorithm. At each iteration, for each image we calculate the overall compatibility values for each the adjacency. We update their compatibility values by increasing support for consistent compatibility values in potential storm tracks and decreasing support for inconsistent compatibility values in potential storm tracks. For each adjacency, we also need to update all its disparities' certainty values in the adjacency. This continues until the difference between the previous certainty and new certainty values of all the disparities converges as specified by a given tolerance threshold, or the maximum number of iterations has been reached.

7.1.1.4 Building Pseudo Tracks

After the relaxation labeling procedure has finished, we use the finalized certainty values of the disparities and the overall compatibility values of adjacencies to build the tracks for storms. In the original storm tracking algorithm, an adjacency has only two disparities, one preceding and one succeeding. By selecting one disparity as one adjacency's succeeding disparity and another adjacency's preceding disparity, we create a chain of disparities as the tracks of storms. Therefore the original storm tracking algorithm only needs to process the disparities in one direction as the image number increase: One disparity is selected as the succeeding disparity of one adjacency. Then we only need to find the right adjacency in the next image that uses it as preceding disparity. However, now in our pseudo tracking algorithm, each adjacency has two sets of disparities instead: the preceding disparity set and the succeeding disparity set. In this case, an adjacency is selected only if all its disparities in both sets are selected, and selecting one disparity in one set of an adjacency cannot guarantee the other disparities from the same set are all selected. Therefore, our new algorithm has to process the selection in two directions for each adjacency, both as the image number is decreasing and as the image number is increasing: when one disparity is selected as one adjacency's either preceding or following disparities, we have to check both sets to make sure all the disparities are successfully selected. The detailed adjacency selection procedure is described as:

1. For the first to the last images in the sequence, find out all the currently available adjacencies with the highest overall compatibility value.
2. Mark this adjacency as “Selected” and then process its two disparity sets; we mark all the disparities in preceding disparity set as “Preceding Selected”; and we mark all the disparities in succeeding disparity set as “Succeeding Selected”.
3. For each disparity in the preceding set, if it hasn’t been marked as ”Succeeding Selected” by another adjacency yet, determine the adjacency with the highest compatibility value that uses this disparity and label it as a succeeding disparity. Repeat step 2 on the newly selected adjacency.
4. For each disparity in the succeeding set, if it hasn’t been marked as ”Preceding Selected” by another adjacency yet, determine the adjacency with the highest compatibility value that uses this disparity and label it as a preceding disparity. Repeat step 2 on the newly selected adjacency.
5. Repeat steps 1 through 4 until there are no available adjacencies left and then continue this processing using the next image, until all the images have been processed.

This advanced pseudo storm tracking algorithm works as a “recursive mutual call” function with adjacency calls to process disparities and disparity calls to process adjacencies (if necessary). An adjacency is selected if and only if all its disparities are still unmarked. This means that if an adjacency uses a disparity in its preceding set, then there are no other adjacencies that have already marked it as a preceding disparity; if an adjacency uses a disparity in its succeeding set, then there are no other adjacencies that already have marked it as a succeeding disparity. Each disparity can only be marked by two adjacencies, one of these adjacencies uses it as one of preceding disparities and the other adjacency uses it as a succeeding disparity. In this way, disparities can be chained together by multiple adjacencies to form the tracks of storms over all the images. The detailed pseudo code functions for selecting tracks through choosing adjacencies and disparities can be found in Algorithms 3, 4 and 5 in Appendix C.

7.1.1.5 Pseudo Storm Tracks Representation

The data structures of our new storm tracking algorithm are different from the original data structures. In the original storm tracking algorithm, all the tracks are single chains connecting storms through disparities. In this case, we just record which storms are in the track and which images they are in. However, in our advanced pseudo storm tracking algorithm, the disparities can merge into a single track and then split into multiple tracks. This produces a complicated network of disparities with "junctions" allowing bifurcation in the storm trajectories. Although there are many methods to represent disparities network, we prefer to use a representation that reduces to the representation used by the original storm tracking algorithm. This keeps the display program downwards compatible with the original storm tracking algorithms. Our representation allows split tracks to be represented by separate curves. We assign an extra **length** value to all the disparities selected for the tracks. The length we use here records how long this disparity is from the last available disparity that it can access. This length value is calculated as in Equation 7.8:

$$depth(\overrightarrow{S_j S_{j+1}}) = \begin{cases} \max(length(\overrightarrow{S_{j+1} S_{j+2}})) + 1, & if\ \overrightarrow{S_{j+1} S_{j+2}}\ exists, \\ 0, & no\ succeeding\ disparity\ exists, \end{cases} \tag{7.8}$$

where $\overrightarrow{S_j S_{j+1}}$ is the current disparity we are investigating and $\overrightarrow{S_{j+1} S_{j+2}}$ represents one of the possible disparities that are connected to this disparity through an adjacency relationship in the succeeding disparity set. The calculation of length begins from the last image in the image sequence and proceeds backwards. Initially, we assign a length value of -1 to all the disparities. Then we assign 0 to those disparities that in tracks but have no succeeding disparities through any adjacency. After these initializations, we calculate the length for the other selected disparities by finding the maximum depth values from all its succeeding disparities and adding 1 to it as the length value for this disparity. In this way, the disparity with the highest length is the one that has the longest track. One example of the depth value calculation can be found in Figure 7.4. This figure shows how the tracks change as time passes and, based on this information, how we can calculate the length value of each disparity. The number marked in red indicates the places where the length is calculated based on a comparison of more than one value of succeeding disparities.

To save these tracks in the same format as used in the original storm tracking algorithm, for each image we choose the one from all the selected disparities with the highest length value and trace it back to the beginning track disparity, recursively using its succeeding disparity until we have reached the last disparity. All these disparities are collected and labelled as "marked" and recorded as belonging to one track. We repeat this process for the other selected disparities, but not for already marked disparities until all the selected disparities are placed in tracks. In this way, all the selected disparities are recorded into separate tracks. The disparities with highest depth values form the longest track. The other tracks could be a totally independent tracks, or part of another track. In order to make the track's merging and splitting run smoothly, we also record each track's origin and destination storms that are parts of other longer tracks. That is, for each track we record an extra two disparities that are connected to the track if applicable. If the track begins from one disparity or ends at another disparity, the track information we record can demonstrate this. After that, we use the same function to display all the tracks, using a cubic β-splines method [56, 57].

7.2 Pseudo Storm Tracking Results and Discussion

We use the same parameters for the pseudo storm tracking algorithm as the original storm tracking algorithm in our experiments so that the results can be compared. The results using the advanced pseudo storm tracking algorithm are shown in the third section of Appendix

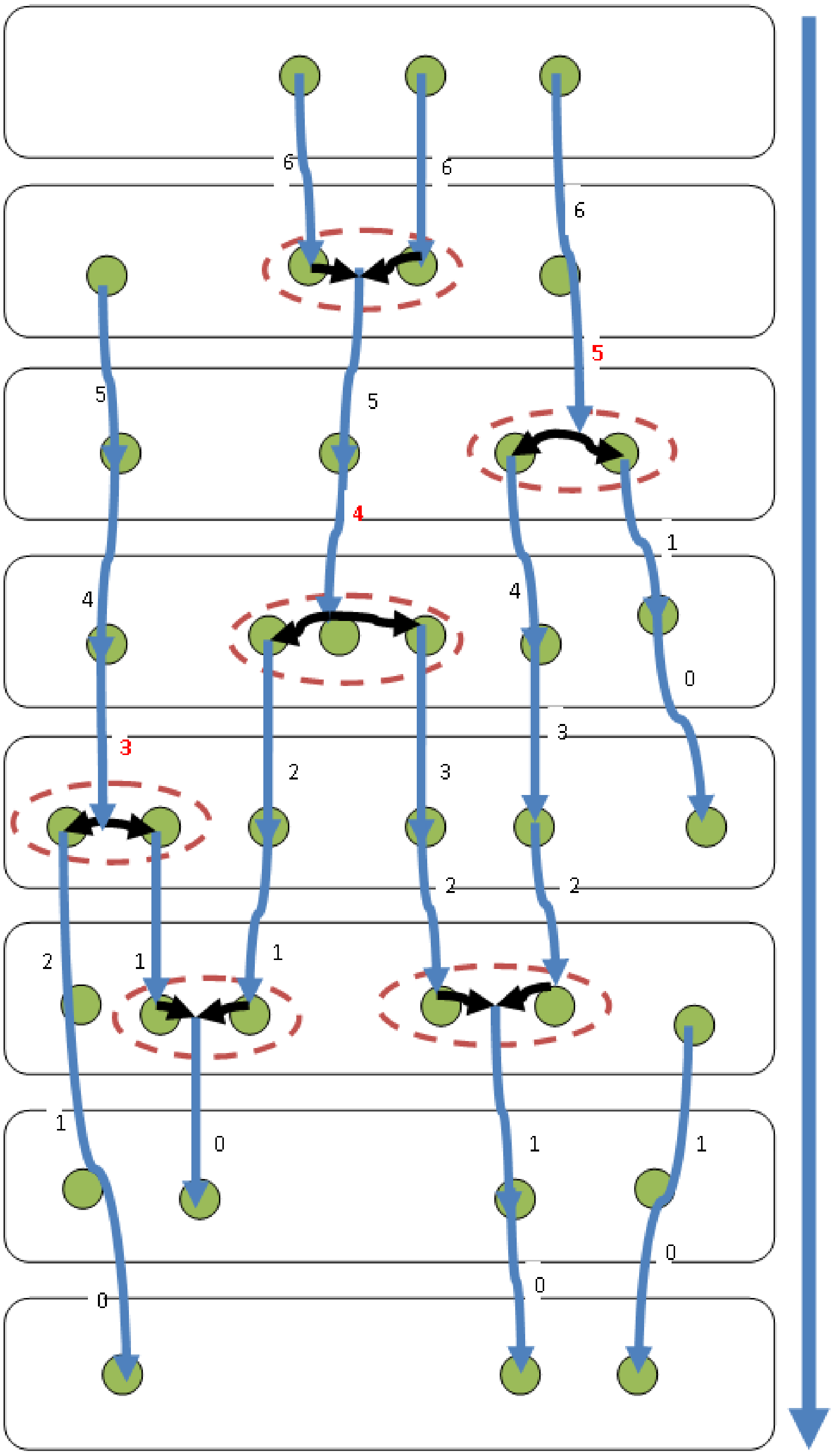


Figure 7.4: The calculation of disparity length among images.

D in Figure D.3. For display purposes, we treat the disparities and connections in the same manner. A connection is considered to be a special disparity connecting two storms in the same image. The tracks are all displayed as generic blue splines. In Figure D.3aa we observe the dark green storm in the middle top part has only one track. In a later image shown in Figure D.3ab, there is now another track detected in this area. In Figure D.3ac these two tracks merge together and, at the same time, another two new tracks are added in this area. These changes indicate that the two real storms detected before have merged into one pseudo storm and then shortly after this merging it splits again into three separate tracks as shown in Figure D.3ae. The three tracks are the track of the tiny orange storm, the track of the red storm, and the tract of huge dark green storm. As shown in later figures, these storms all become smaller and smaller and eventually split into still smaller storms until all the storms disappear, as shown in Figure D.3al.

Another major storm track is the track in the middle bottom part of the images. Figure D.3aa shows the two bigger storms, the dark blue one and the light green one, as first having their own tracks at the beginning but being connected later when the storms merge. The two separate tracks continue until Figure D.3ax, where a pseudo storm first appears. The two real storms get merged into this pseudo storm in this image. These two real storms are tracked together in later images as a pseudo storm constantly until a split occurs in Figure D.3ax. The steady track shows us how the huge storm has moved as time elapsed, and the red spot indicates the current position of the storm along the steady track. These two storms are then tracked separately as shown in Figures D.3az to D.3ba. None of the other tracks have a pseudo storm and remain the same as in the original storm tracking algorithm (as they should).

Appendix E shows another group of tracking results with the same two Doppler radars for another time period. The results are collected between 8:00 to 14:00 on August 19th, 2007. We track the storms over this 6 hour period with both the original and pseudo storm tracking algorithms. We observed the life cycle of a huge severe weather storm; how it was formed, lasted for a couple of hours and then disappeared. The storms there move east, and are more isolated than in the previous sequences, with less merging and splitting. The performances of two methods are very straightforward. The same conclusion can be made that the advanced pseudo storm tracking algorithm provides more stable and reasonable tracking results.

7.3 Conclusions

All our recovered tracks qualitatively well represent the trajectories of storms by taking the storm events (storm shape and size changes and storm merging and splitting) into account. Our advanced pseudo storm tracking algorithm outperforms the original storm tracking algorithm by using pseudo storms when storm merging or splitting occur. Now storm merging and splitting are explicitly and correctly accounted for. Pseudo storms definitely add value to the tracking process.

Chapter 8

Conclusion and Future Works

8.1 Contributions and Conclusions

It has been two decades since our research group (the first MSc thesis was by Zhang [82] in 1991) began research of Doppler radar imagery using Computer Vision optical flow and tracking methods. That research mainly focused on two aspects: how to compute full velocity using radial velocity information and how to detect and track severe weather storms using the reflectivity data. the storm detection and tracking using reflectivity data.

In this thesis, our main contributions are:

1. We extended the original 3D optical flow least squares and regularization algorithms to multiple overlapping Doppler radars. Our framework allows a network of several Doppler radars that cover overlapping areas to cooperate with each other and track storms across radar coverage areas. The number of overlapping radars is allowed to be any number, so potentially a huge network of radars covering the whole of North America could be used. We mainly focused on the multiple Doppler radars around the Great Lakes area (this meant our experimental analysis was limited to the dual radar case). We designed synthetic Doppler data and quantitatively analyzed our algorithms using this. We also qualitatively analyzed our real radar data. From our synthetic experiments, we conclude that the dual (multiple) Doppler radar scheme is

capable of generating more stable and more reliable results, especially in the common coverage areas, than a single radar can. The overall output error, magnitude error and direction error all decreased impressively compared to the single radar case. Our analysis of the three velocity components shows the difficulties in recovering velocity information on the vertical axes. This difficult arises due to the well known aperture problem. This quantitative analysis allowed use to choose good thresholds for the qualitative analysis on the real Great Lakes data.

2. In order to enhance the performance of our algorithm on the velocity retrieval in the vertical direction (our z dimension), we used data from a windprofiler radar in conjunction with data from an overlapping Doppler. We presented algorithms that showed how to combine or integrate these data and achieve higher accuracy in the velocity components along the vertical axis. Again, we demonstrated our algorithm on both synthetic (quantitatively) and real data (qualitatively). In both cases, our algorithm produced more accurate optical flow fields in the vicinity of the windprofiler radar. We also investigated the effect of various parameter values on the full velocity and separately on each of the x, y and z component velocities for the synthetic data. By examining the performance on synthetic data, we showed that quantitatively more accurate and detailed information could be recovered along the z (height) dimension and also in the x and y dimensions. However, it should be noted that our windprofiler radar only covered a small part of the Doppler radar coverage area. This limits its application (we need a large network of windprofiler radars overlapping with one of more Doppler radars to see its real power). Our framework is able to handle many windprofiler radars but we do not yet have data from multiple overlapping windprofiler and Doppler radars. The strategic placement of windprofiler radars, for example, near airports or cities, might also increase the usefulness of our recovered wind velocity information. We also test the similar scheme using radiosonde data, same results are concluded.

3. We also investigated severe storm detection and tracking using an already developed relaxation labeling algorithm, but using multiple radar data. Due to the overlap of several radars on the Canadian and US sides of the border and the data format change from NEXRAD *I* to NEXRAD *II*, tracking in the Great Lakes area become complicated. We restricted our radar data to be that available on the NCDC (National Climate Data Center) website[1]. One of the compatibility functions in our relaxation labeling tracking algorithm was velocity compatibility but, for the reasons outlined in Chapter 6, we did not use velocity compatibility in our new tracking algorithm.

4. We expanded on a new concept of "Pseudo Storm" (proposed by Krezeski *et al.* in 1994 [41]) to handle the problems of tracking non-rigid storms, that sometimes changed their size and shape significantly between adjacent images, sometimes merged with other neighbouring storms into a bigger storm or sometimes split into a number of smaller storms. First, we presented the definition and computational algorithm for pseudo storms, which were used to represent collections of smaller storms as a bigger storm for tracking with another larger storm. Second, we modified the original tracking algorithm to use restricted pseudo storms in it. It turns out that by adding pseudo storms to the original tracking algorithm the results are much smoother, but the tracks are also shorter since the current tracking algorithm cannot recognize the connection between pseudo storm and the real storms that comprise it. Thirdly, we redesigned the tracking algorithm by adding "n-to-1" relationship to the disparity structure and adjacency structure to handle pseudo storms thoroughly. The relaxation labeling algorithm was also modified appropriately. The advanced pseudo storm tracking algorithm is backwards compatible with the original tracking algorithm, in that the same tracks were obtained for any data satisfying the original algorithm's assumptions. We compared real data storm trajectories using this pseudo storm tracking algorithm with the original storm tracking algorithms. Our new pseudo storm tracking algorithm provided much better trajectories. especially

---

[1] `http://www.ncdc.noaa.gov/`

when complicated storm events, like merging or splitting ones, occurred.

8.2 Future Research

Our ultimate objective is to combine all the available radars covering the entire Lake Erie area, i.e. the NCDC NEXRAD *II* radars at Detroit (Michigan), Cleveland (Ohio) and Buffalo (New York), the Canadian Doppler radars at King City and Exeter and the Canadian windprofiler radars at Harrow and Walsingham, comprising part of the O-Q net network that overlaps these Doppler radars, and obtain and analyze this data simultaneously to compute high quality 3D wind velocities everywhere in real time. Such a analysis tool would allow us to perform a more complete investigation of severe weather events over this region, such as long term tracking of storms using the pseudo storm concept. Thus, we could issue warnings for incoming severe storms promptly and we could obtain a better understanding of a severe storm's life cycle.

There is a lot of further work that needs to be done for the tracking algorithm. We need to optimize this algorithm so it can be used on real data in real time. All our processing is currently offline using the whole volume of images together at one time. Therefore it will be interesting to find out the performance of our algorithm working in the real-time environment.

Of course, we do not want to limit ourselves to the Great Lakes area: detecting and tracking storms in the many overlapping Doppler radars along the eastern coast of the US or in the southern US would also greatly interest us. Lastly, the regularization framework is capable of integrating any other velocity constraint from other sources, for example, radiometer data [8].

We have developed useful and accurate optical flow and storm tracking algorithms to aid in the meteorological understanding of storms and their nowcasting/forecasting.

Bibliography

[1] L. Armijo. A theory for the determination of wind and precipitation velocities with Doppler radars. *Journal of Atmospheric Sciences*, 26:570–5732, 1968.

[2] M. E. Baldwin. Verification of the time evolution of precipitation systems in numerical weather forecasts. In *2005 IEEE International Conference on Robotics and Automation*, pages 1276–1281. IEEE, 2005.

[3] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12:43–77, 1994.

[4] J. L. Barron, R. E. Mercer, D. Cheng, and P. Joe. Tracking 'fuzzy' storms in Doppler radar images. *Computer Vision and Applications Handbook*, pages 807–820, 1999.

[5] J. L. Barron, R. E. Mercer, D. Cheng, and P. Joe. *Tracking 'Fuzzy' Storms in Doppler Radar Images*, pages 626–627. Computer Vision and Applications Handbook: A Guide for Students and Practitioners, 2000.

[6] J. L. Barron, R.E. Mercer, X. Chen, and P. Joe. 3D velocity from 3D Doppler radial velocity. *Intl. J. of Imaging Systems and Technology*, 15:189–198, 2005.

[7] J.L. Barron. Experience with 3D optical flow on gated MRI cardiac datasets. In 1^{st} *Canadian Conference on Computer and Robot Vision (CRV2004)*, pages 370–377, May 2004.

[8] L. Bianco, D. Cimini, F. S. Marzano, and R. Ware. Combining microwave radiometer and wind profiler radar measurements for high-resolution atmospheric humidity profiling. *Journal of Atomsphere and Occeanic Technology*, 22:949–965, 2005.

[9] D. J. Boccipio. A diagnostic analysis of the VVP single-Doppler retrieval technique. *Journal of Atmospheric and Occeanic Technology*, 12:230–248, 1995.

[10] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):1–21, 2005.

[11] D. Caya and I. Zawadzki. VAD analysis of nonlinear wind fields. *Journal of Atmospheric and Occeanic Technology*, 9:575–587, 1992.

[12] K. Chaudhury, R. Mehrota, and C. Srinivasan. Detecting 3d flow. In *IEEE International Conference*, volume 2, pages 1073–1078. Robotics and Automation, May 1994.

[13] X. Chen. 3D velocity from Doppler radial velocity. Master's thesis, University of Western Ontario, London, Ontario, March 2001.

[14] X. Chen, J. L. Barron, and R. E. Mercer. 3D regularized velocity from 3D Doppler radial velocity. In *International Conference on Image Processing*, volume 3, pages 664–667, Thessaloniki, Greece, 2001.

[15] X. Chen, J. L. Barron, R. E. Mercer, and P. Joe. 3D least squares velocity from 3D doppler radial velocity. In *Vision Interface*, pages 55–63, Ottawa, 2001.

[16] D. Cheng. Tracking fuzzy storm centers in Doppler radar images. Master's thesis, Dept. of Computer Science, University of Western Ontario, London, Ontario, Canada, June 1996.

[17] D. Cheng, R. E. Mercer, J. L. Barron, and P. Joe. Tracking fuzzy storm centers in Doppler radar images. In *IEEE International Conference on Image Processing*, volume 2, pages 959–962, 1996.

[18] M. Chong and J. Testud. Three-dimensional wind field analysis from dual-Doppler radar data. Part III: Boundary condition: An optimum determination based on variational concept. *Journal of Climate and Applied Meteorology*, 22:1227–1241, 1983.

[19] M. Chong, J. Testud, and F. Roux. Three-dimensional wind field analysis from dual-Doppler radar data. Part II: Minimizing the error due to temporal variation. *Journal of Climate and Applied Meteorology*, 22:1216–1226, 1983.

[20] R. Cifelli, S. A. Rutledge, D. J. Boccippio, and T. Matejka. Horizontal divergence and vertical velocity retrivals from Doppler radar and wind profiler observations. *Journal of Atmospheric and Oceanic Technology*, 13:948–966, 1996.

[21] G. P. Cressman. An operational objective analysis system. *Monthly Weather Review*, 87:367–374, 1959.

[22] M. Dixon and G. Wiener. Titan: Thunderstorm identication, tracking, analysis and nowcasting-a radar-based methodology. *Journal of Atmospheric and Oceanic Technology*, 10:785–795, 1993.

[23] Christian Doppler. Über das farbige licht der doppelsterne (concerning the coloured light of double stars), May 1842. Presentation at Royal Bohemian Society of Sciences.

[24] R. J. Doviak and P. S. Ray. Error estimation in wind fields derived from dual-Doppler radar measurement. *Journal of Applied Meteorology*, 15:868–878, 1975.

[25] R. J. Doviak and D. S. Zrnić. *Doppler Radar and Weather Observations*. Incorporation of Academic Press, Orlando, Florida, United States of America, 1984.

[26] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2001.

[27] C. C. Easterbrook. Estimating horizontal wind fields by two-dimensional curve fitting of single Doppler radar measurements. In *16th Radar Meteorology Conference*, pages 214–219, 1975.

[28] W. L. Ecklund, D. A. Carter, and B. B. Balsley. A UHF wind profiler for the boundary layer: Brief description and initial results. *Journal of Atomsphere and Occeanic Technology*, 5:432–441, 1988.

[29] K. S. Gage, C. R. Williams, and W. L. Ecklund. UHF wind profilers: A new tool for diagnosing tropical convective cloud systems. *Bulletin of the American Meteorological Society*, 75:2289–2294, 1994.

[30] T. Gal-Chen. Errors in fixed and moving frames of references: Applications for conventional and Doppler radar analysis. *Journal of Atmospheric Sciences*, 39:2279–2300, 1982.

[31] J. Gao, M. Xue, A. Shapiro, and K. K. Droegemeier. A variational method for the analysis of three-dimensional wind fields from two Doppler radars. *Monthly Weather Review*, 127:2128–2142, 1998.

[32] J. Gong, L. Wang, and Q. Xu. A three-step dealiasing method for doppler velocity data quality control. *Journal of Atmospheric and Oceanic Technology*, 20:1738–1748, 2003.

[33] M. A. Greminger and B. J. Nelson. A deformable object tracking algorithm robust to occlusions and spurious edges. In *2005 IEEE International Conference on Robotics and Automation*, pages 1276–1281. IEEE, 2005.

[34] W. K. Hocking. System design, signal-processing and preliminary results for the Canadian (London, Ontario) VHF atmospheric radar. *Radio Science*, 11:687–706, 1997.

[35] W. K. Hocking, P. A. Taylor, P. S. Argall, I. Zawdzki, F. Fabry, G. McBean, R. Sica, H. Hangan, G. Klaassen, J. Barron, and R. Mercer. A VHF windprofiler network in ontario and quebec, canada: Design details and capabilities. In *Proceedings of the 33rd Conference on Radar Meteorology*, pages 13B.1, 7pp., Cairns, Queensland, Australia, August 2007. American Meteorological Society.

[36] W. K. Hocking, P. A. Taylor, I. Zawdzki, P. S. Argall, F. Fabry, G. McBean, R. Sica, H. Hangan, G. Klaassen, J. Barron, and R. Mercer. A new 40-55 MHz windprofiler network in ontario and quebec, canada - the o-qnet. In *Proc. of the Eleventh International Workshop on Technical and Scientific Aspects of MST Radar Gadanki*, pages 377–380, Macmillan, India, December 2006.

[37] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–204, 1981.

[38] J. T. Johnson, P. L. Mackeen, A. Witt, E. D. Mitchell, G. J. Stumpf, M. D. Eilts, and K. W. Thomas. The storm cell identication and tracking algorithm: An enhanced wsr-88d algorithm. *Weather Forecasting*, 13:263–276, 1998.

[39] R. Klette, K. Schluns, and A. Koschan. *Computer Vision: Three Dimensional Data from Images*, chapter 5. Springer, 1998.

[40] A. J. Koscielny, R. J. Doviak, and R. Rabin. Statistical considerations in the estimation of divergence from single-Doppler radar and application to prestorm boundary-layer observations. *Journal of Applied Meteorology*, 21:197–210, 1982.

[41] D. Krezeski, R. E. mercer, J. L. Barron, P. Joe, and H. Zhang. Storm tracking in Doppler radar images. In *IEEE Proceedings on Image Processing*, pages 226–230, 1994.

[42] H. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[43] S. Laroche and I. Zawadzki. A variational analysis method for retrieval of three-dimentional wind field from single Doppler radar data. *Journal of the Atmospheric Science*, 51:2664–2682, 1994.

[44] S. Laroche and I. Zawadzki. Retrievals of horizontal winds from single Doppler clear-air data by methods of cross correlation and variational analysis. *Journal of the Atmospheric and Oceanic Technology*, 12:721–738, 1995.

[45] R. M. Lhermitte and D. Altas. Precipitation motion by pulse Doppler. In *Ninth Weather Radar Conference*, pages 218–223, 1961.

[46] M. Lothon, B. Campistron, S. Jacoby-Koaly, B. Benech, F. Lohou, and F. Girard-Ardhuin. Comparison of radar reflectivity and vertical velocity observed with a scannable c-band radar and two UHF profilers in the lower troposphere. *Journal of Atmospheric and Occeanic Technology*, 19:899–910, 2002.

[47] M. Lothon, B. Campistron, S. Jacoby-Koaly, B. Benech, F. Lohou, F. Girard-Ardhuin, and A.Druilhet. Reply on comments of radar reflectivity and vertical velocity observed with a scannable c-band radar and two UHF profilers in the lower troposphere. *Journal of Atmospheric and Occeanic Technology*, 20:1224–1229, 2003.

[48] B.D. Lucas and T. Kanade. An iterative image-regression technique with an application to stereo vision. *DARPA Image Understanding Workshop*, pages 121–130, 1981.

[49] C. Lucas, P. T. May, and R. A. Vincent. An algorithm for the detection of fronts in wind profiler data. *Weather and Forecasting*, 16:234–247, 2001.

[50] H. Maisel and Giuliano. *Simulation of Discrete Stochastic Systems*, pages 51–52, 153–154. Science Research Associates, 1972.

[51] P. T. May and D. K. Rajopadhyaya. Wind profiler observations of vertical motion and precipitation microphysics of a tropical squall line. *Monthly Weather Review*, 124:621–633, 1996.

[52] P. T. May and R. G. Strauch. An examination of wind profiler signal processing algorithms. *American Meteorological Society*, 6:731–736, 1989.

[53] R. E. Mercer, J. L. Barron, and D. Cheng. Fuzzy points: Algebra and application. *Pattern Recognition*, 35:1153–1166, 2002.

[54] J. J. O'Brien. Alternative solutions to the classical vertical velocity problem. *Journal of Applied Meteorology*, 9:197–203, 1970.

[55] C. Qiu and Q. Xu. A simple adjoint method of wind analysis for single-Doppler data. *Journal of the Atmospheric and Oceanic Technology*, 9:588–598, 1992.

[56] W. Qiu. 3D Doppler storm tracking. Master's thesis, University of Western Ontario, London, Ontario, April 2001.

[57] W. Qiu, R. E. Mercer, J. L. Barron, and P. Joe. 3D storm tracking in 3D Doppler precipitation reflectivity datasets. In *Irish Machine Vision and Image Processing Conference*, pages 79–86, 2001.

[58] P. S. Ray, R. J. Doviak, G. B. Walker, D. Sirmans, J. Carter, and B. Bumgarner. Dual-Doppler observation of a tornadic storm. *Journal of Applied Meteorology*, 14:1521–1530, 1975.

[59] R. E. Rinehart. *Radar for Meteorologists*. Rinehart Publications, Grand Forks, United States of America, 1999.

[60] N. Sengupta and R. J. Watson. Detection of ducts by wind profiler radar. In *Twelfth International Conference*, volume 1, pages 59–62. Antennas and Propagation, 2003.

[61] A. Shapiro, S. Ellis, and J. Shaw. Single-Doppler velocity retrievals with phoenix ii data clear air and microburst wind retrievals in the planetary boundary layer. *Journal of the Atmospheric Science*, 52:1265–1287, 1995.

[62] A. Shapiro and J. J. Mewes. New formulations of dual-Doppler wind analysis. *Journal of Atmospheric and Oceanic Technology*, 16:782–792, 1998.

[63] A. Shapiro, P. Robinson, J. Wurman, and J. Gao. Single-Doppler velocity retrieval with rapid-scan radar data. *Journal of the Atmospheric and Oceanic Technology*, 20:1758–1775, 2003.

[64] H. Spies, H. Haußecker, B. Jähne, and J.L.Barron. Differential range flow estimation. In *21.Symposium fur Mustererkennung (DAGM1999)*, pages 309–316. Springer, Bonn, Sept. 1999.

[65] H. Spies, B. Jähne, and J. L. Barron. Range flow estimation. *Computer Vision Image Understanding*, 85(3i):209–231, March 2002.

[66] C. B. Storlie, T. C. M. Lee, J. Hanning, and D. Nychka. Tracking of multiple merging and splitting targets: a statistical perspective. *Statistica Sinica*, 19:1–52, 2009.

[67] R. G. Strauch, B. L. Weberand A. S. Frisch, C. G. Little, D. A. Merritt, K. P. Moran, and D. C. Welsh. The precision and relative accuracy of profiler wind measurements. *Journal of Atmospheric and Oceanic Technology*, 4:563–571, 1987.

[68] X. Tang. Tracking 3D Doppler weather storms using fuzzy ellipsoids and radial velocity. Master's thesis, University of Western Ontario, London, Ontario, January 2003.

[69] X. Tang, J. L. Barron, R. E. Mercer, and P. Joe. Tracking weather storms using 3D doppler radial velocity information. In *13th Scandinavian Conference on Image Analysis*, pages 1038–1043, 2003.

[70] X. Tang, R. E. Mercer, J. L. Barron, and P. Joe. Tracking 3D storms using fuzzy points represented as ellipsoids. In *Irish Machine Vision and Image Processing Conference*, pages 73–82, 2003.

[71] J. Testud and M. Chong. Three-dimensional wind field analysis from dual-Doppler radar data. Part I: Filtering, interpolating and differentiating the raw data. *Journal of Climate and Applied Meteorology*, 22:1204–1215, 1983.

[72] P. Waldteufel and H. Corbin. On the analysis of single-Doppler radar data. *Journal of Applied Meteorology*, 18:532–542, 1979.

[73] R. M. Worthington. Comments on comparison of radar reflectivity and vertical velocity observed with a scannable c-band radar and two UHF profilers in the lower troposphere. *Journal of Atmospheric and Occeanic Technology*, 20:1221–1229, 2003.

[74] D. B. Wuertz, B. L. Weber, R. G. Strauch, A. S. Frisch, C. G. Little, D. A. Merritt, K. P. Moran, and D. C. Welsh. Effects of precipitation on UHF wind profiler measurements. *Journal of Atmosphere and Occeanic Technology*, 5:450–465, 1988.

[75] G. Xu and V. Chandrasekar. Radar storm motion estimation and beyond: A spectral algorithm and radar observation based dynamic model. In *World Weather Research Programme Symposium on Nowcasting and Very Short Range Forecasting*, 2005.

[76] Q. Xu, K. Nai, and L. Wei. An innovation method for estimating radar radial-velocity observation error and background wind error covariances. *Quarterly Journal of The Royal Meteorological Society*, 133:407–415, 2006.

[77] Q. Xu, K. Nai, and L. Wei. Fitting vad winds to aliased doppler radial-velocity observations: A global minimization problem in the presence of multiple local minima. *Q. J. R. Meteorol. Soc.*, 136:451–461, 2009.

[78] Q. Xu, K. Nai, L. Wei, P. Zhang, and S. Lin. A vad-based dealiasing method for radar velocity data quality control. *Journal of Atmospheric and Oceanic Technology*, 28:50–62, 2011.

[79] Q. Xu, K. Nai, L. Wei, and Q. Zhao. An unconventional approach for assimilating aliased radar radial velocities. *Tellus*, 61A:621–630, 2009.

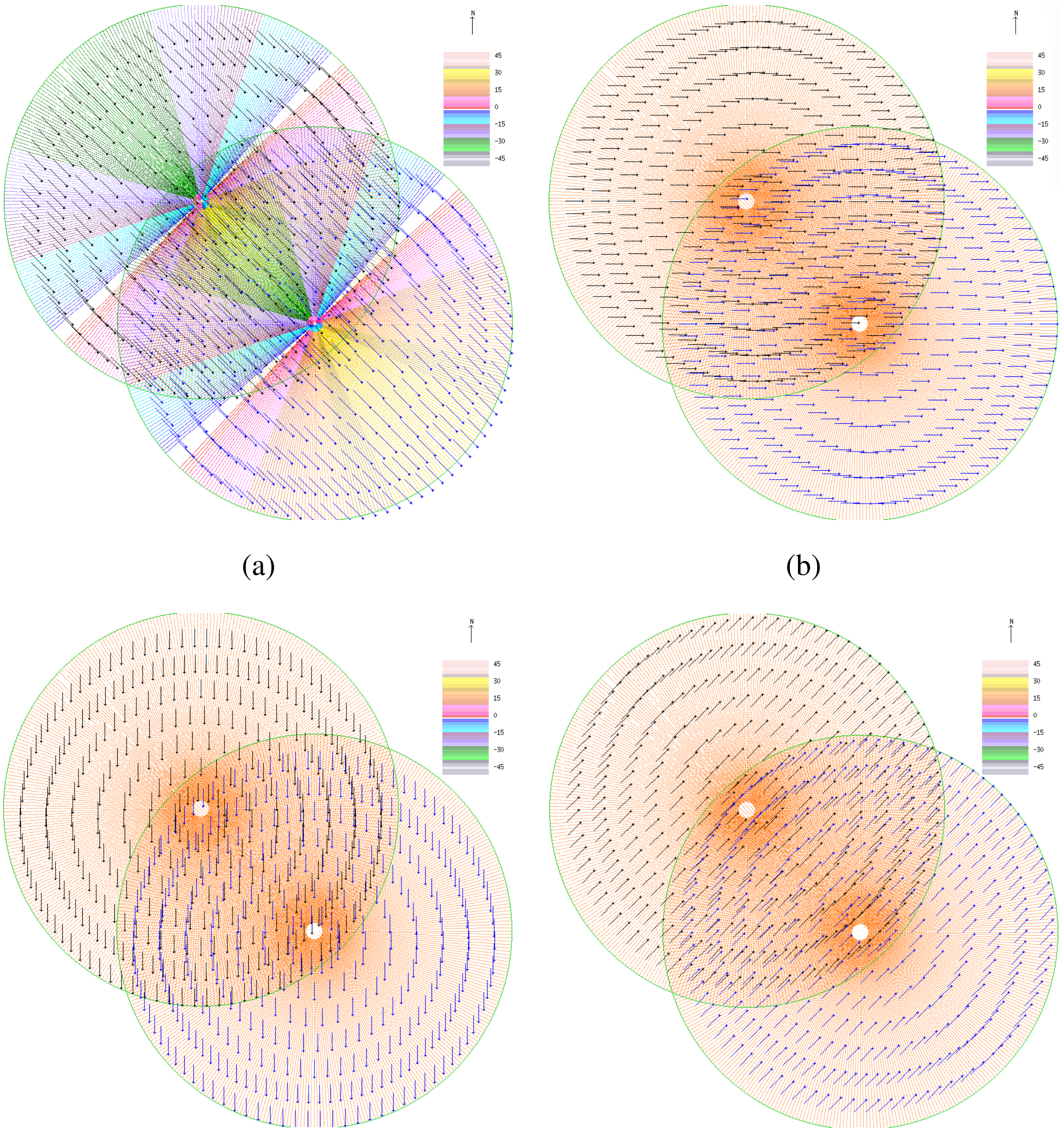
[80] A. Yilmaz, O. Javed, and M Shah. Object tracking: A survey. *ACM Computing Surveys*, 38, 2006.

[81] J. G. Yoe, M. F. Larsen, and E. J. Zipser. VHF wind profiler data quality and comparison of methods for deducing horizontal and vertical air motions in a mesoscale convective storm. *Journal of Atmospheric and Oceanic Technology*, 9:713–727, 1992.

[82] H. Zhang. Storm detection in radar images. Master's thesis, Dept. of Computer Science, University of Western Ontario, London, Ontario, Canada, July 1991.

Appendix A

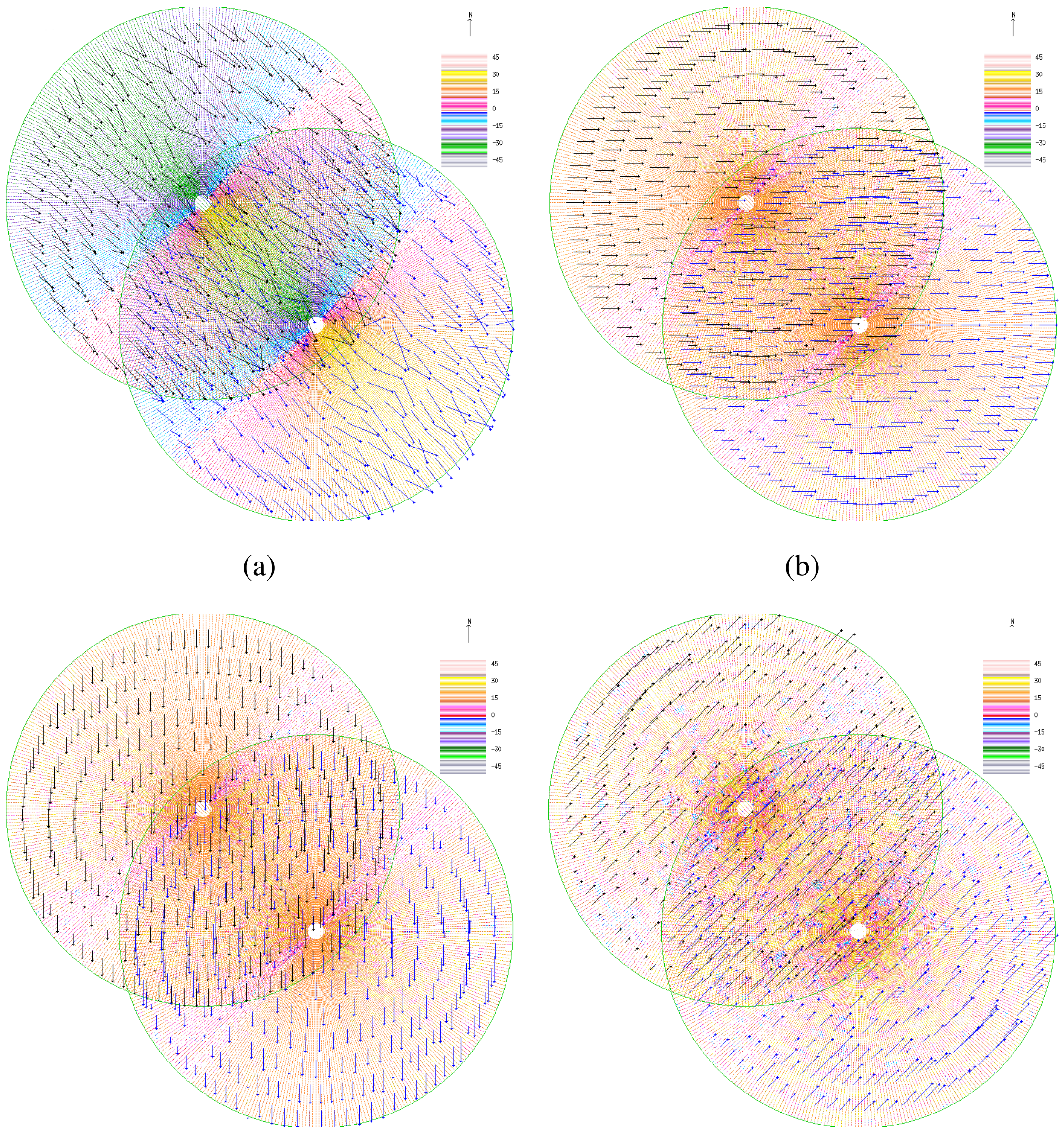
Synthetic Experiment Results for the Least Squares Method



(a) (b)

(c) (d)

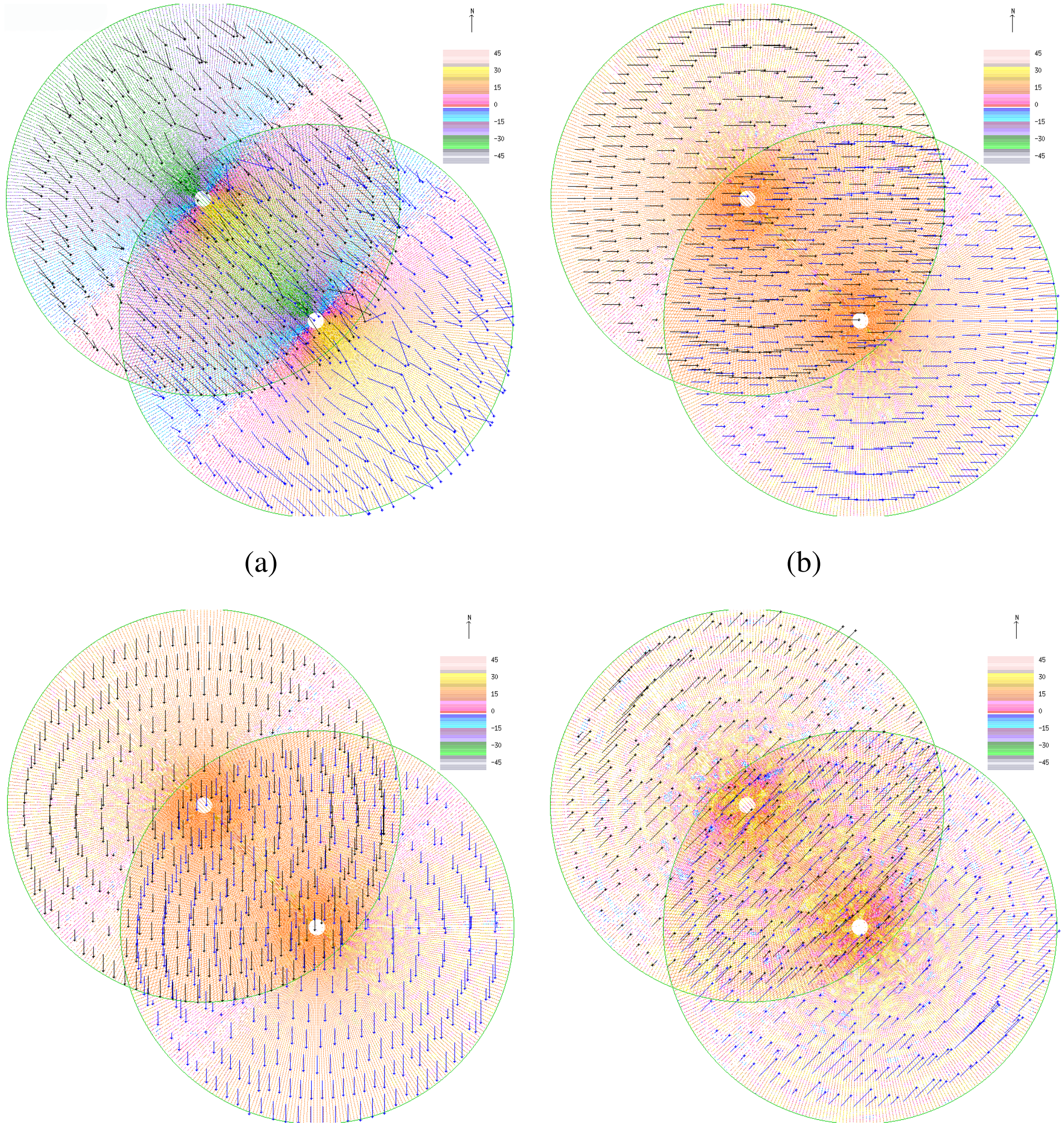
Figure A.1: The correct synthetic velocity of group 1 at variation level $K = 0$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

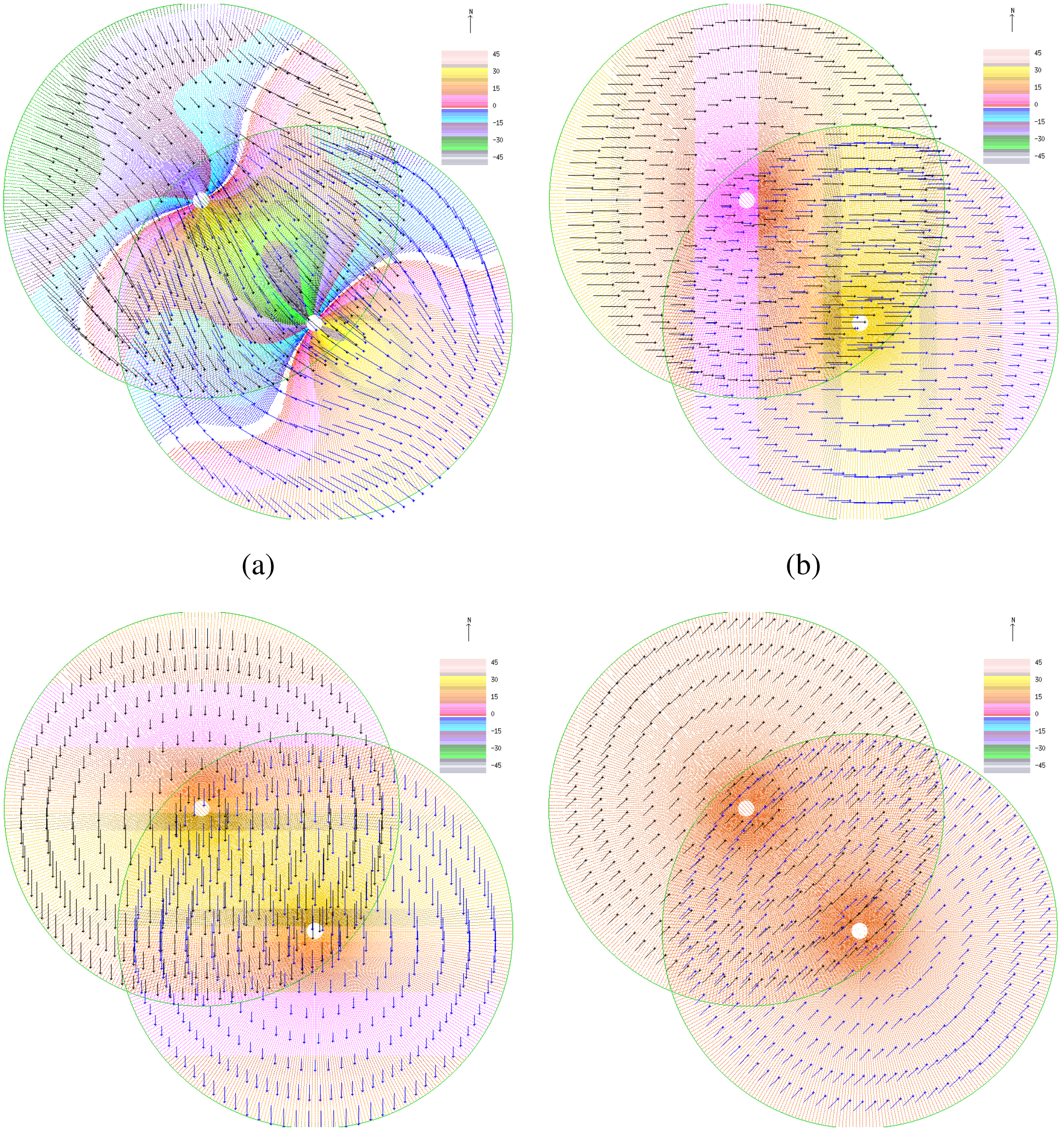
Figure A.2: The single retrieved synthetic velocity of group 1 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

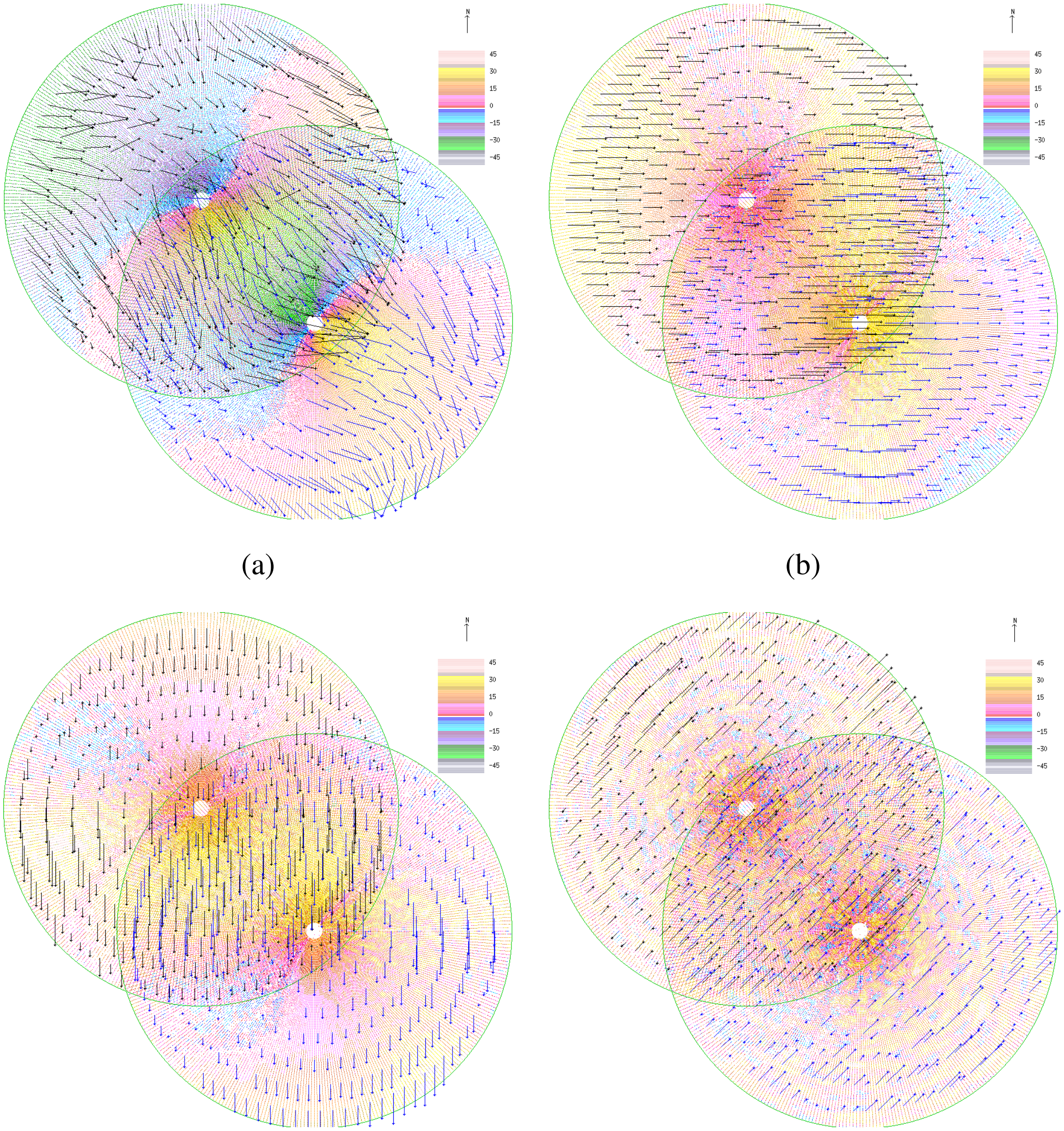
Figure A.3: The dual retrieved synthetic velocity of group 1 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

Figure A.4: The correct synthetic velocity of group 1 at variation level $K = 5$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a)

(b)

(c)

(d)

Figure A.5: The single retrieved synthetic velocity of group 1 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.

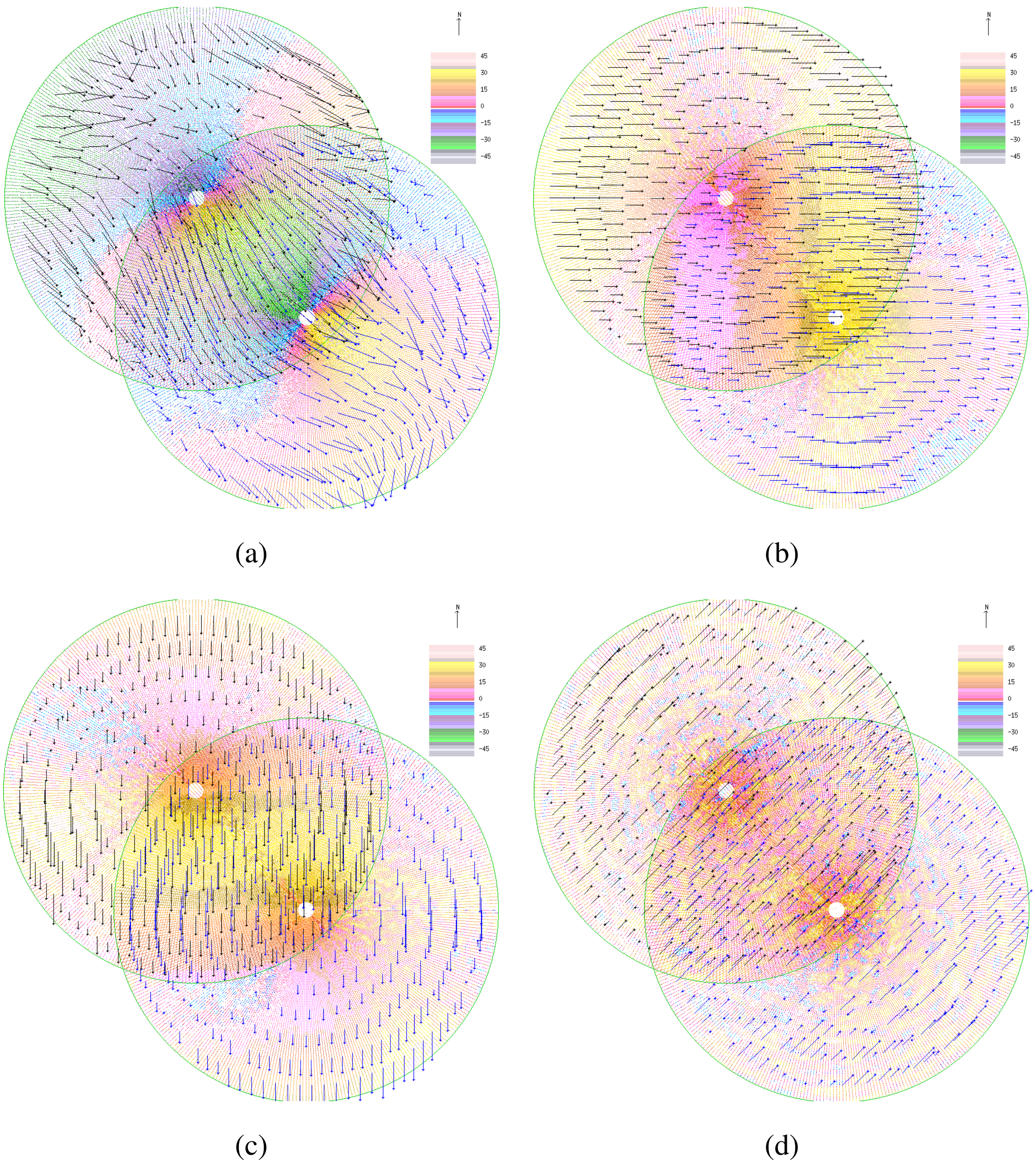
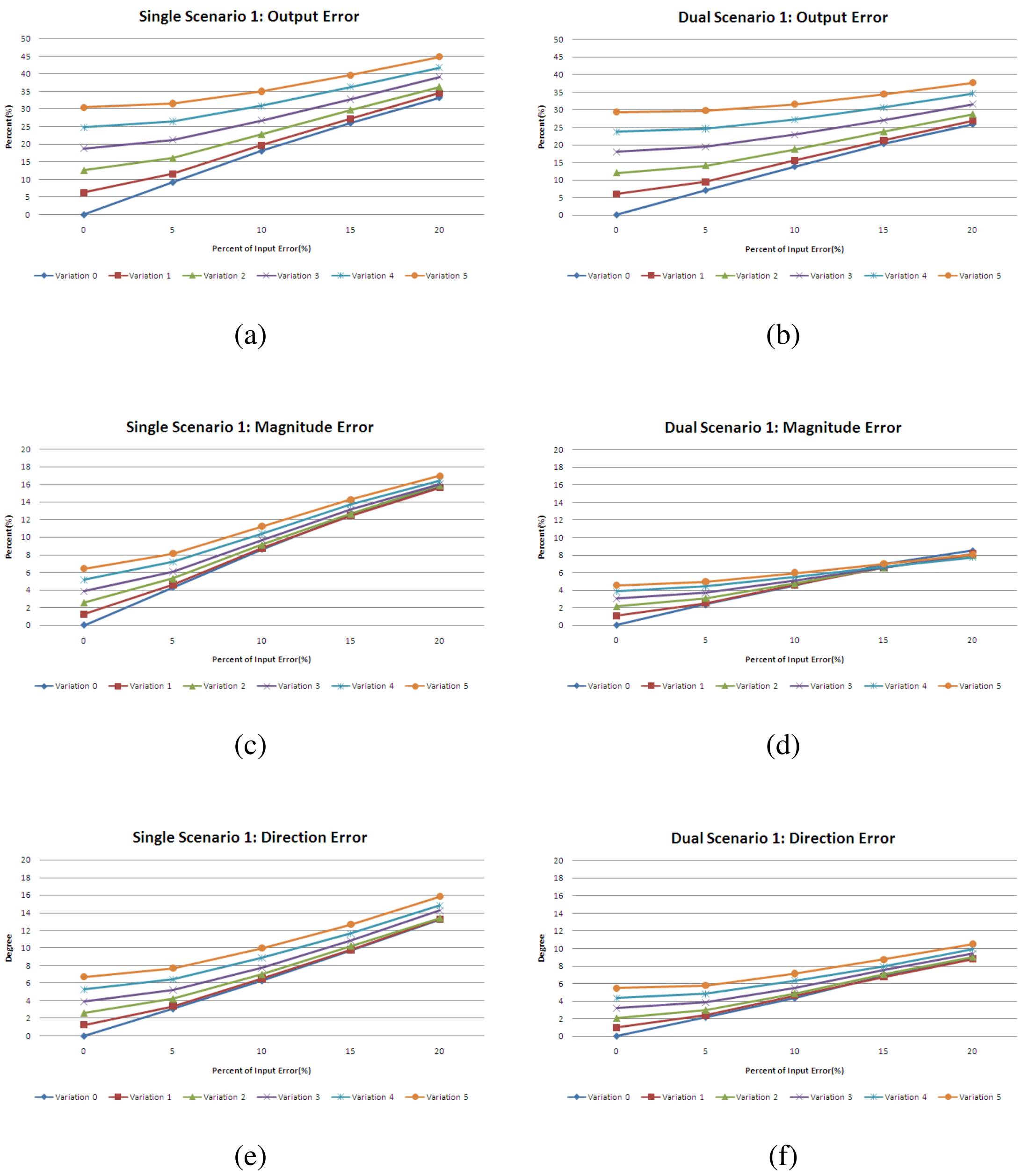


Figure A.6: The dual retrieved synthetic velocity of group 1 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.

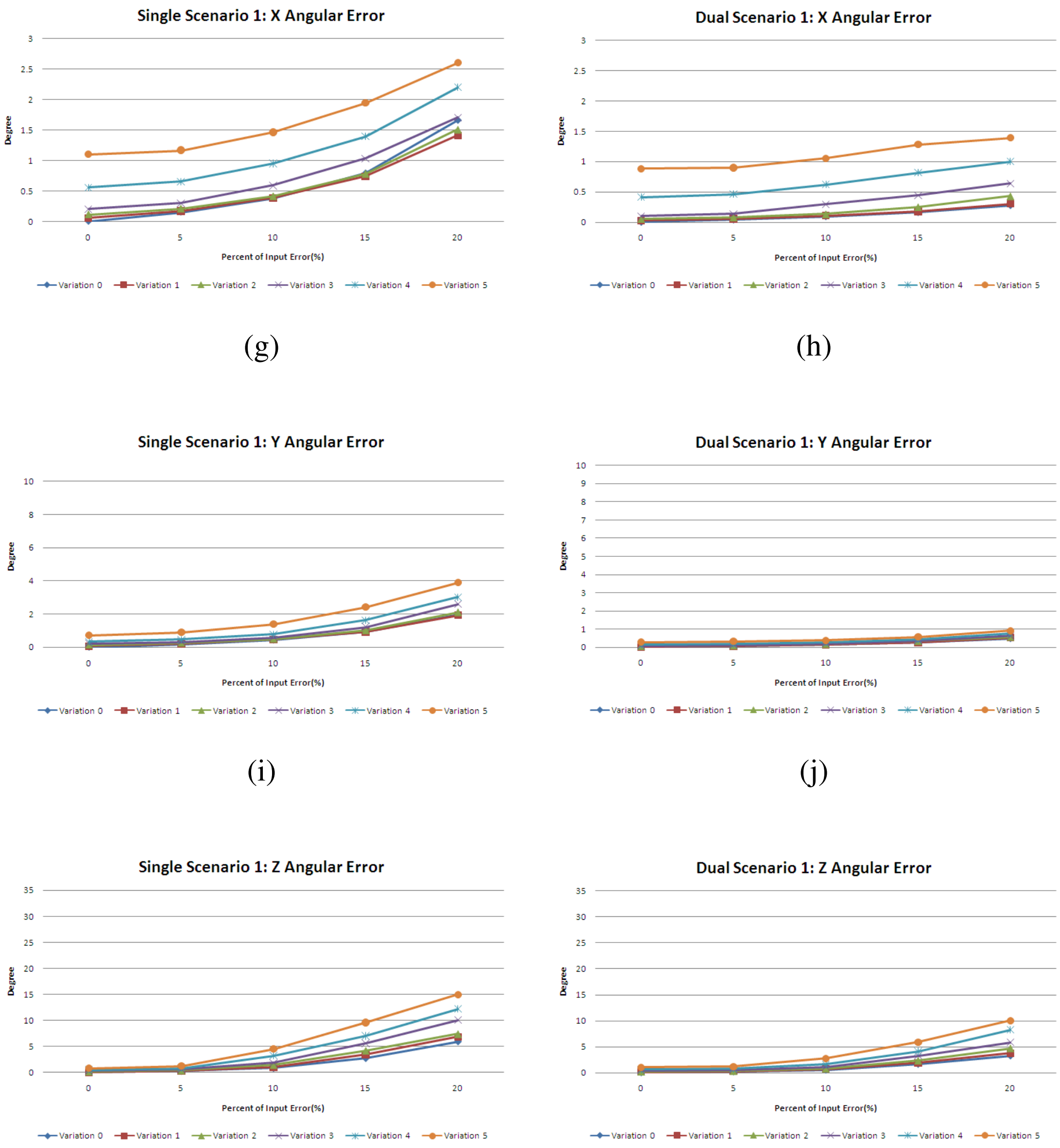


(a) (b)

(c) (d)

(e) (f)

Figure A.7: Result Analysis of Synthetic Data Group 1: (20.0, 20.0, 20.0). The Single Retrieval (a), (c), (e) and Dual Retrieval (b), (d), (f).

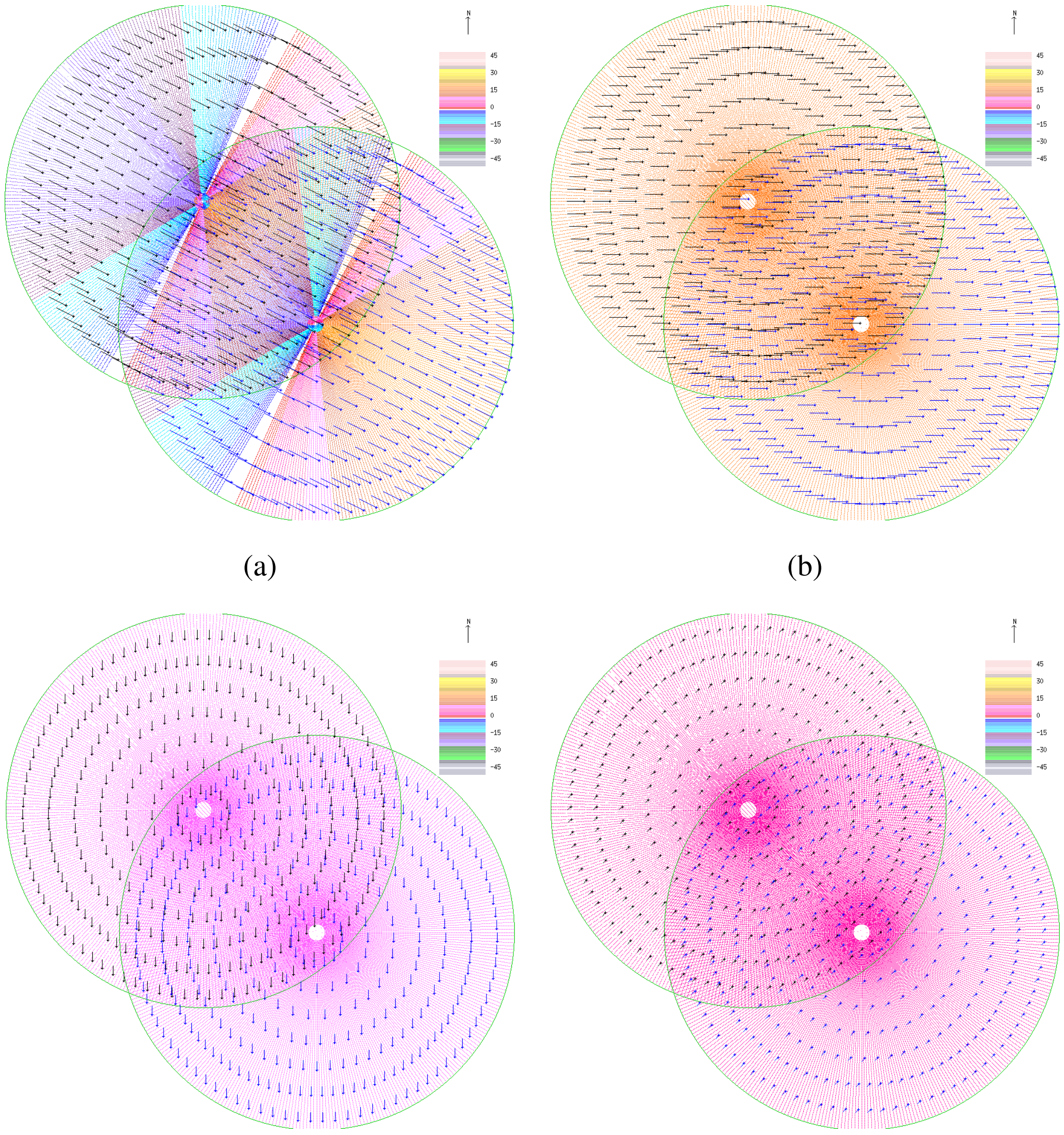


(g) (h)

(i) (j)

(k) (l)

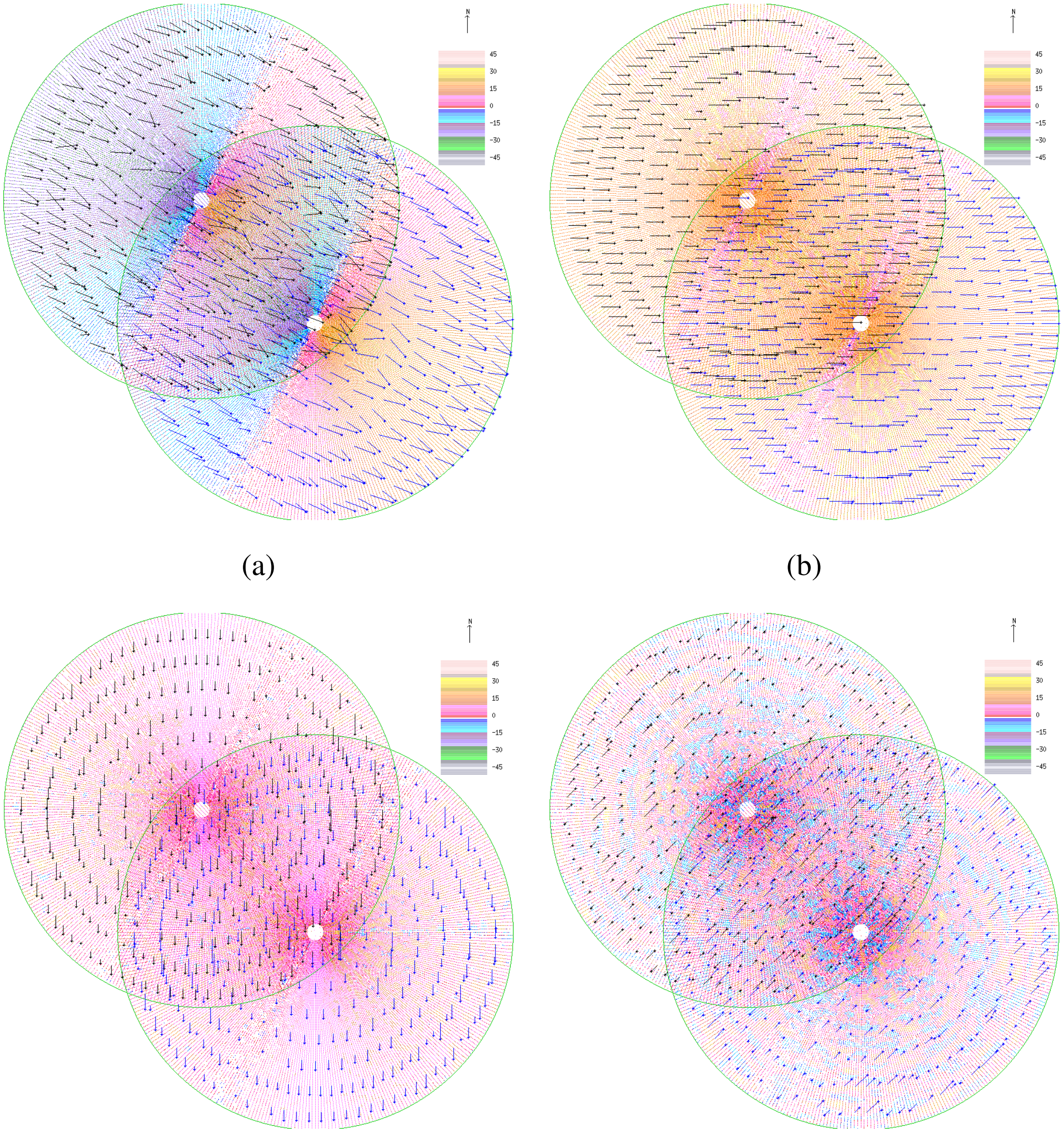
Figure A.7: Result Analysis of Synthetic Data Group 1: (20.0, 20.0, 20.0). The Single Retrieval (g), (i), (k) and Dual retrieval (h), (j), (l).



(a) (b)

(c) (d)

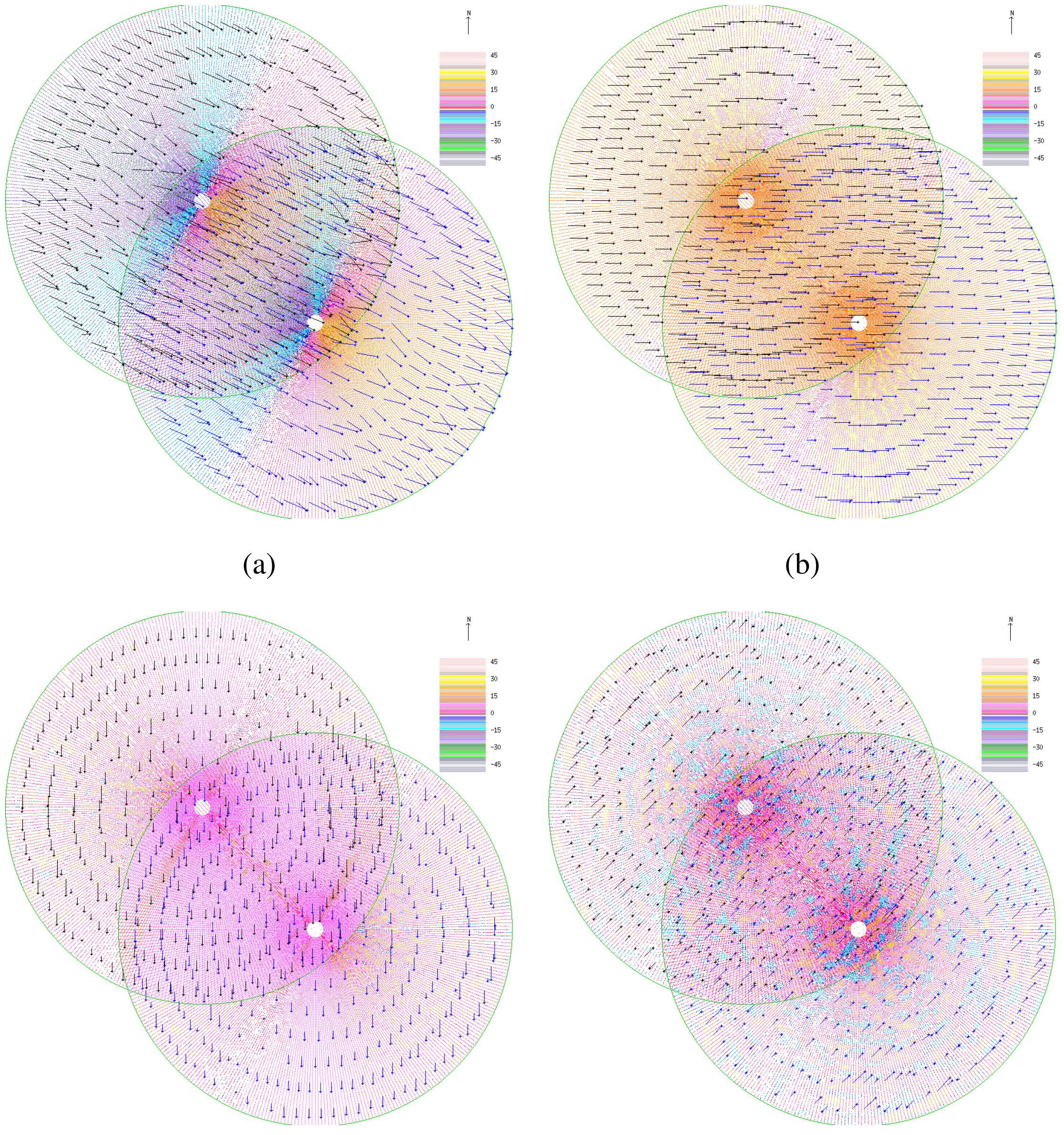
Figure A.8: The correct synthetic velocity of group 2 at variation level $K = 0$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

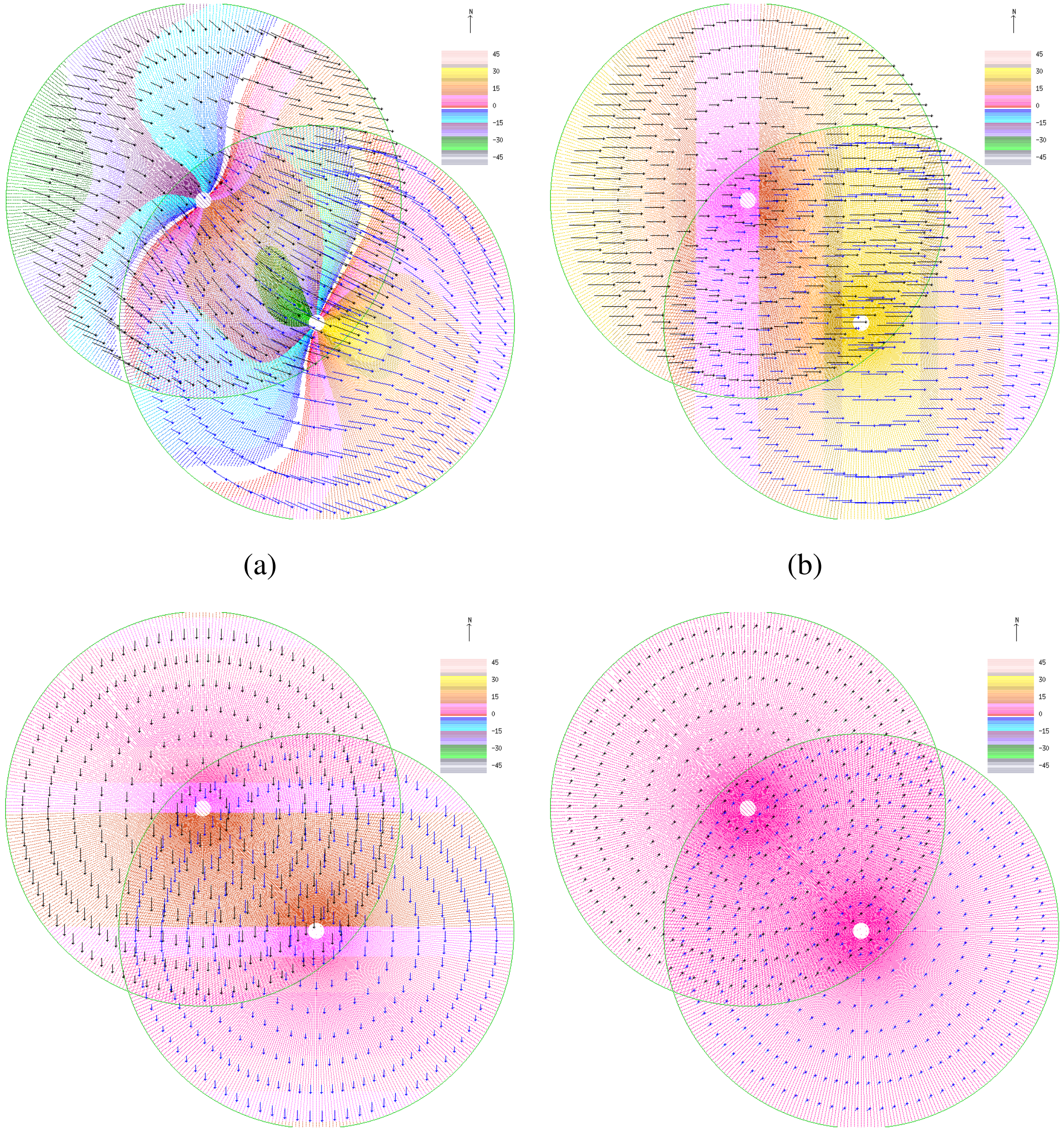
Figure A.9: The single retrieved synthetic velocity of group 2 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

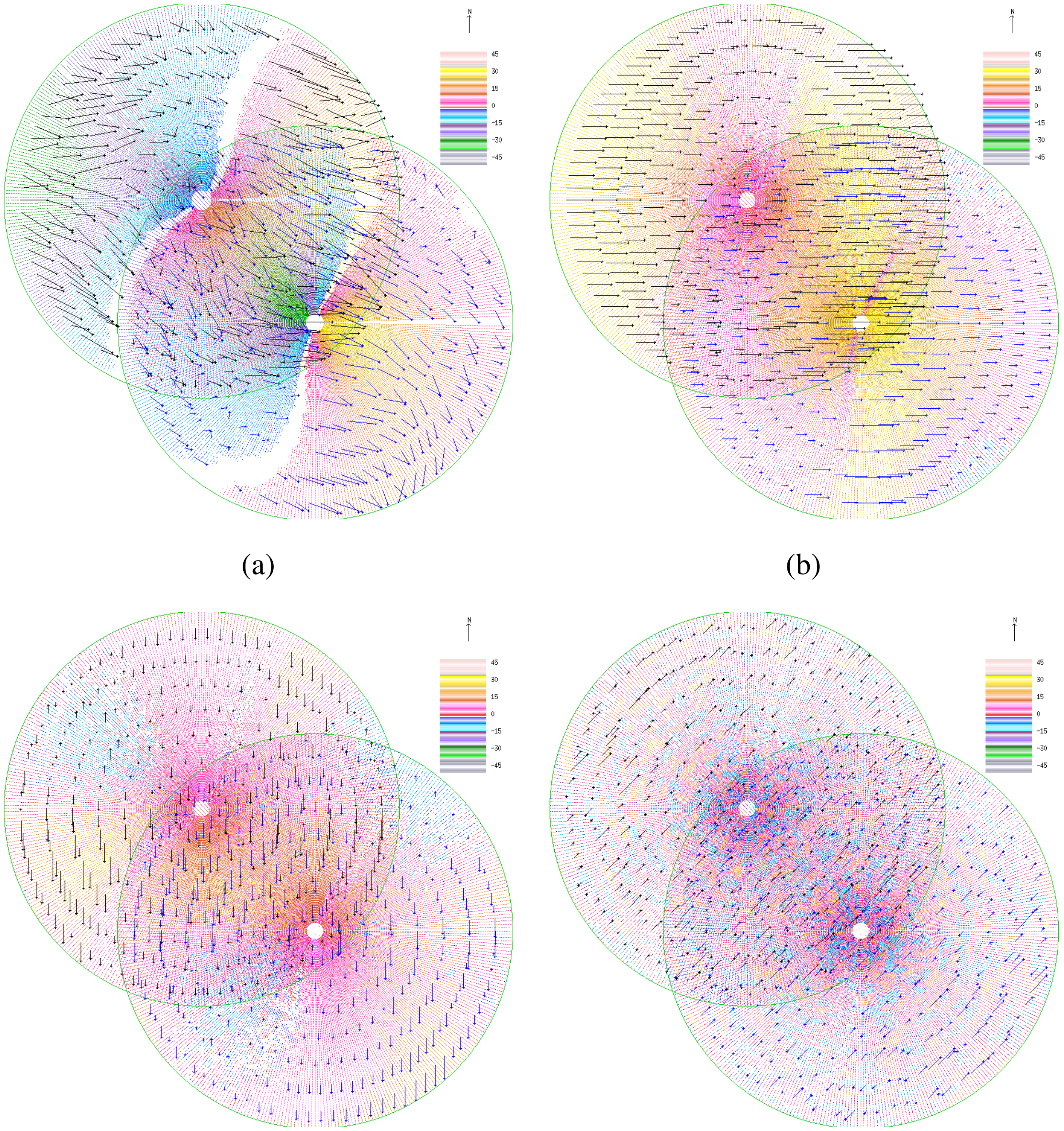
Figure A.10: The dual retrieved synthetic velocity of group 2 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

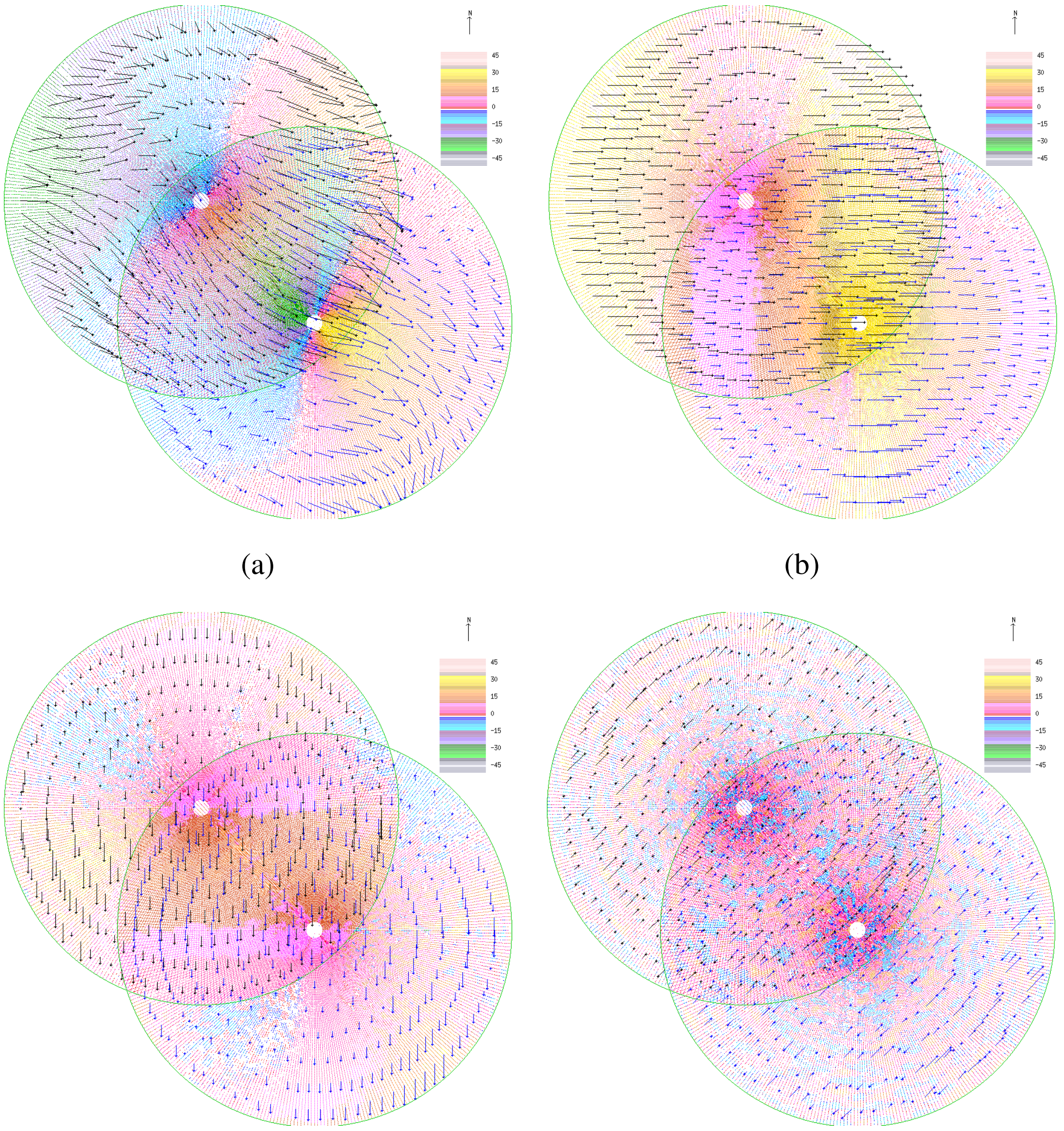
Figure A.11: The correct synthetic velocity of group 2 at variation level $K = 5$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

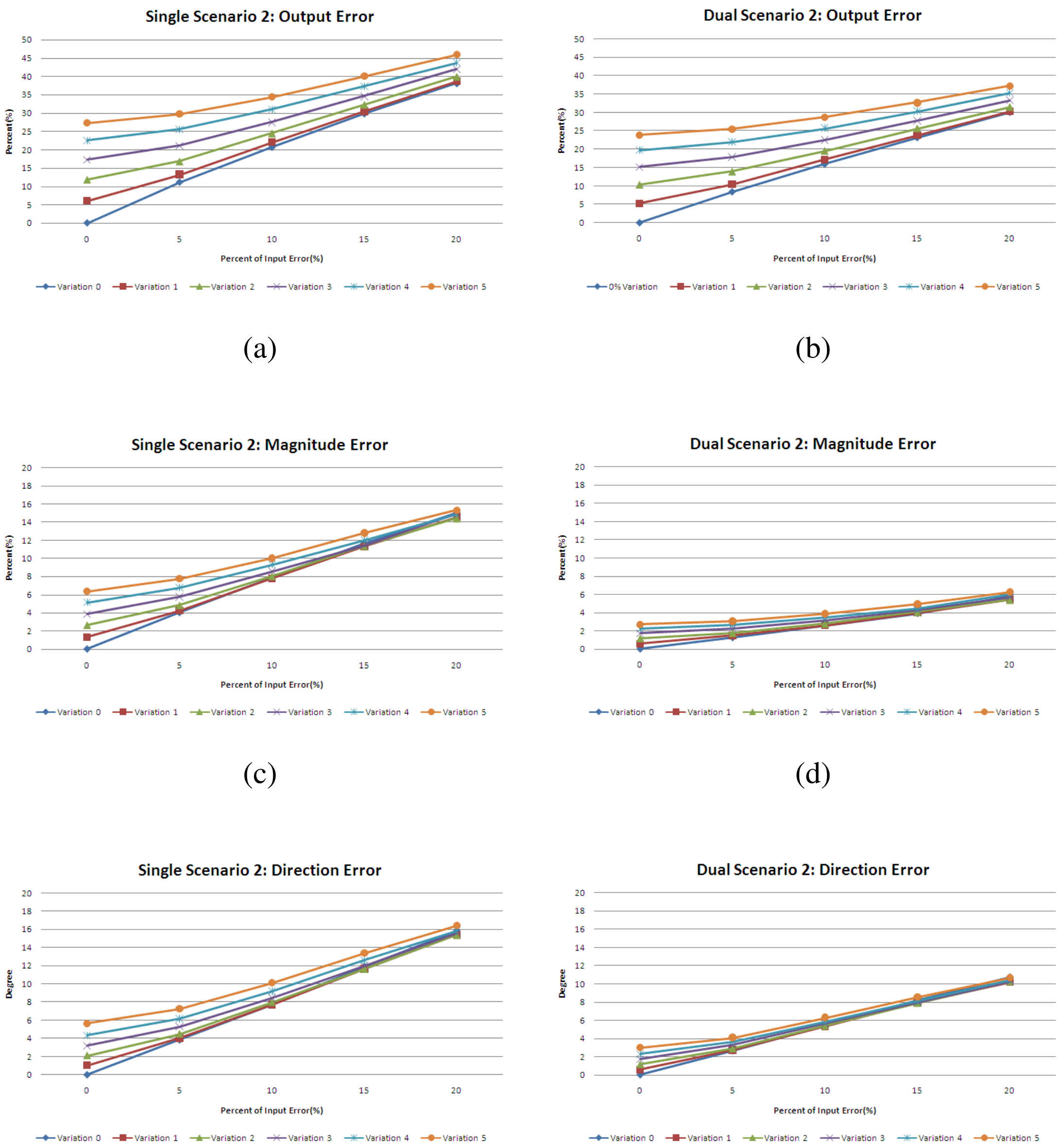
Figure A.12: The single retrieved synthetic velocity of group 2 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

Figure A.13: The dual retrieved synthetic velocity of group 2 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.

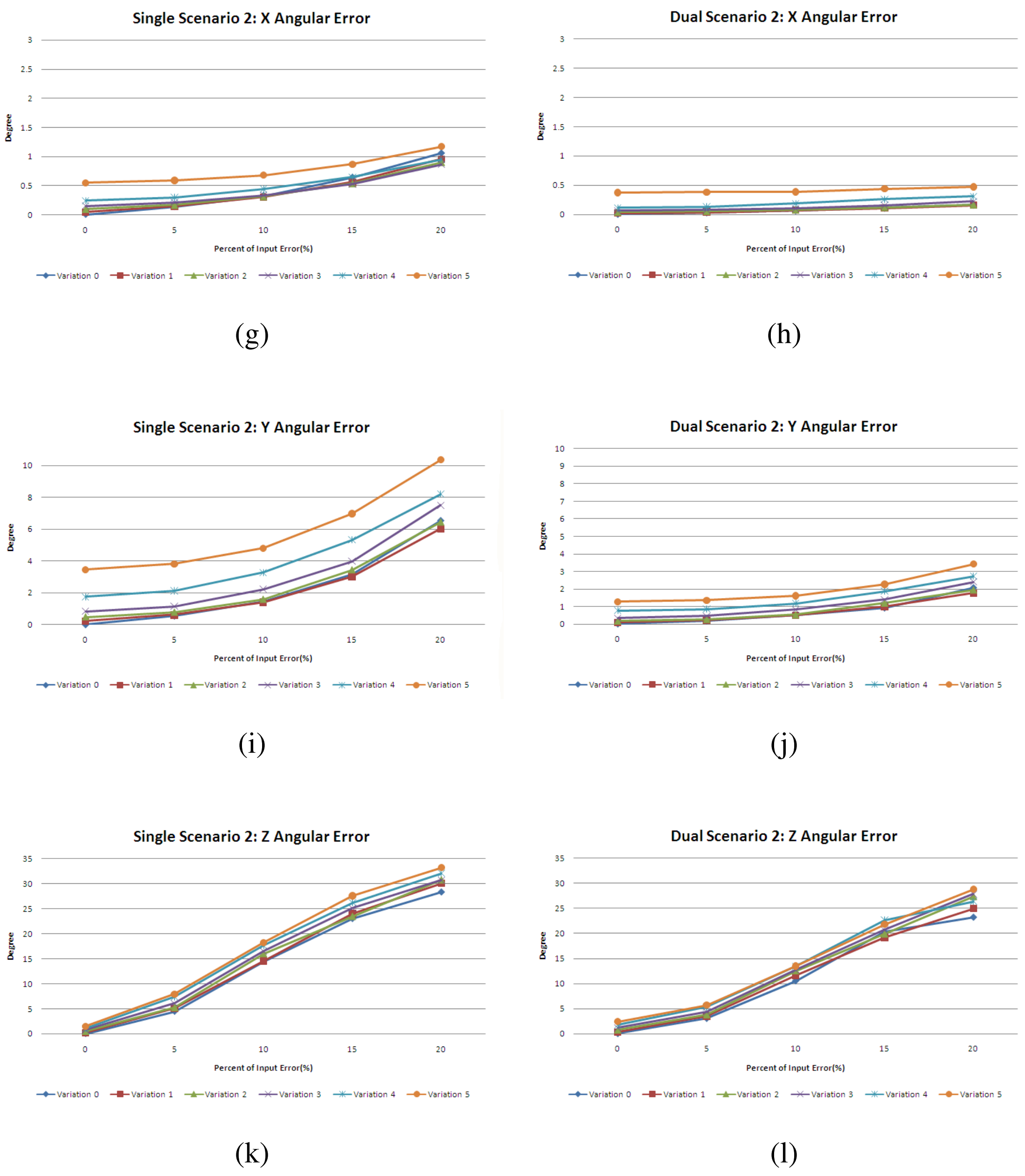


(a) (b)

(c) (d)

(e) (f)

Figure A.14: Result Analysis of Synthetic Data Group 2: (20.0, 10.0, 5.0). The Single Retrieval (a), (c), (e) and Dual Retrieval (b), (d), (f).

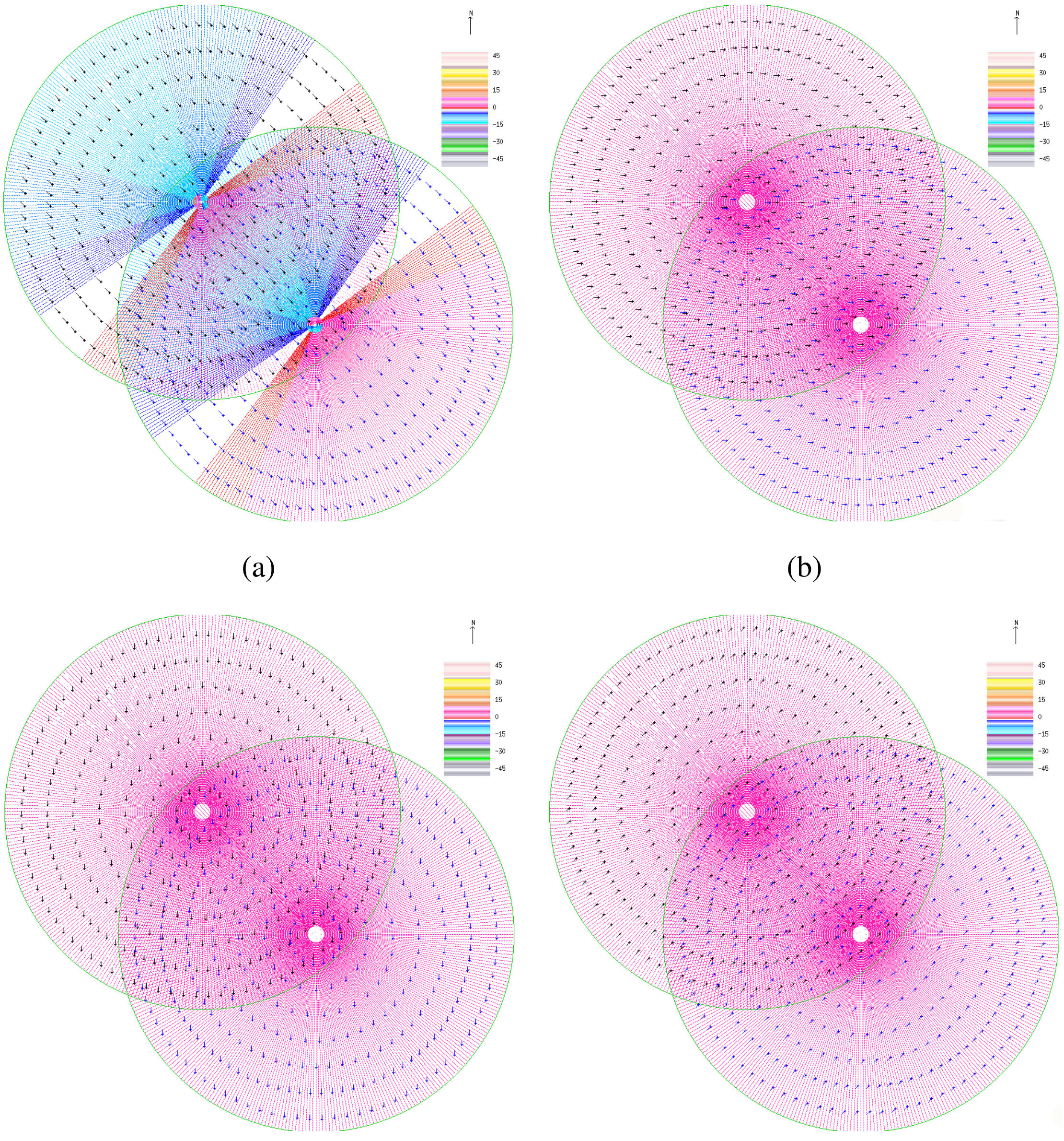


(g) (h)

(i) (j)

(k) (l)

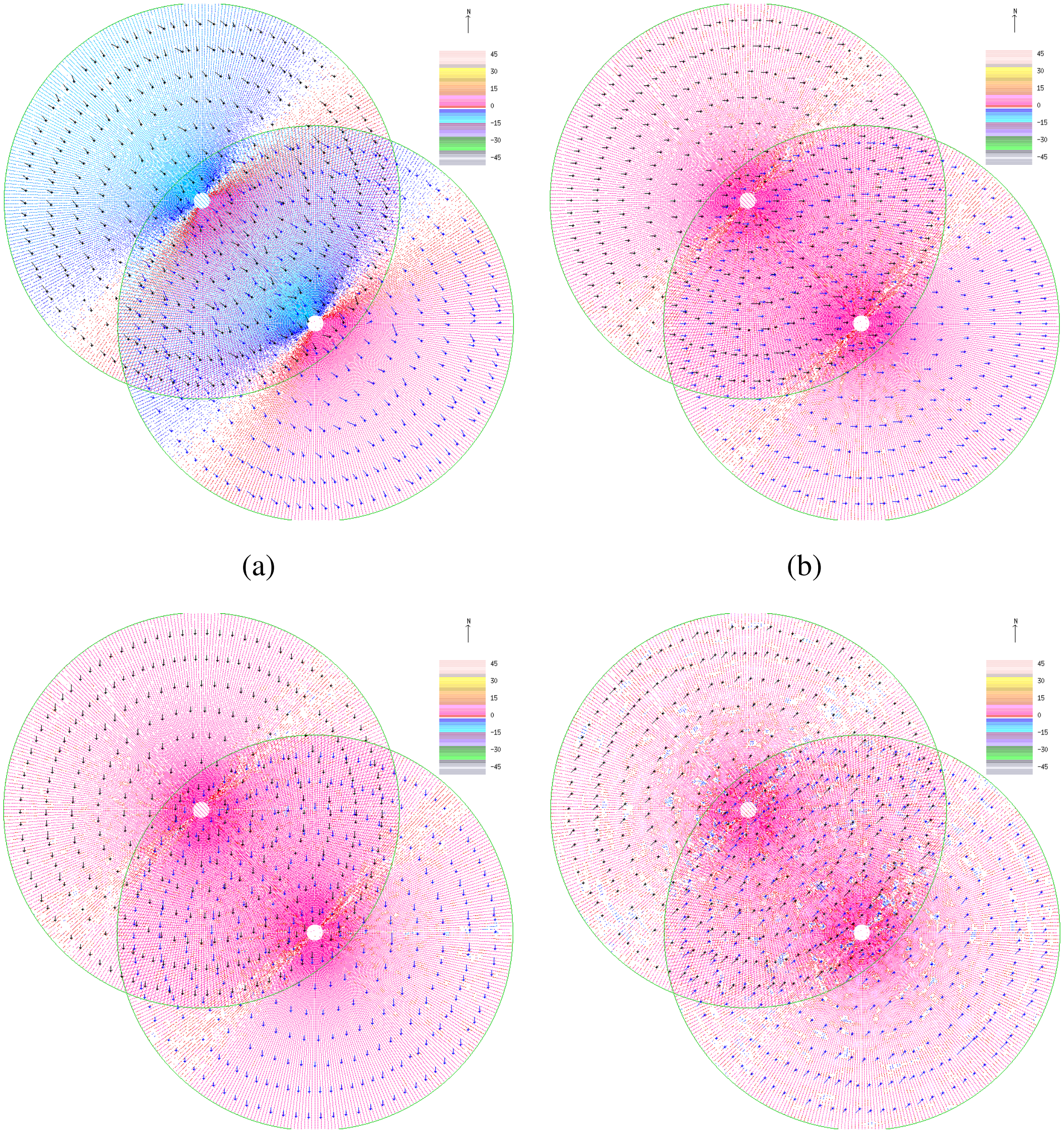
Figure A.14: Result Analysis of Synthetic Data Group 2: (20.0, 10.0, 5.0). The Single Retrieval (g), (i), (k) and Dual retrieval (h), (j), (l).



(a) (b)

(c) (d)

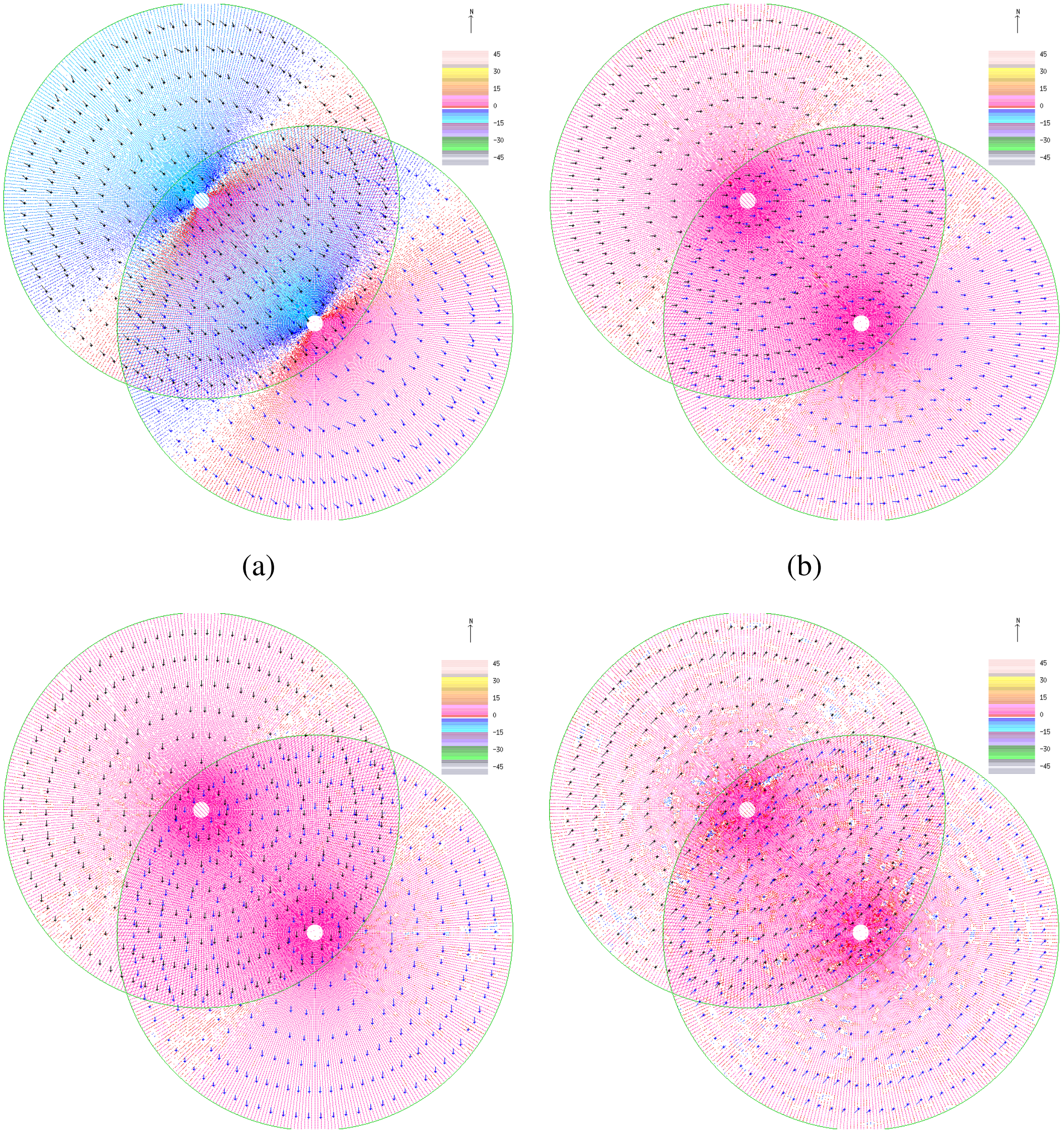
Figure A.15: The correct synthetic velocity of group 3 at variation level $K = 0$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

Figure A.16: The single retrieved synthetic velocity of group 3 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



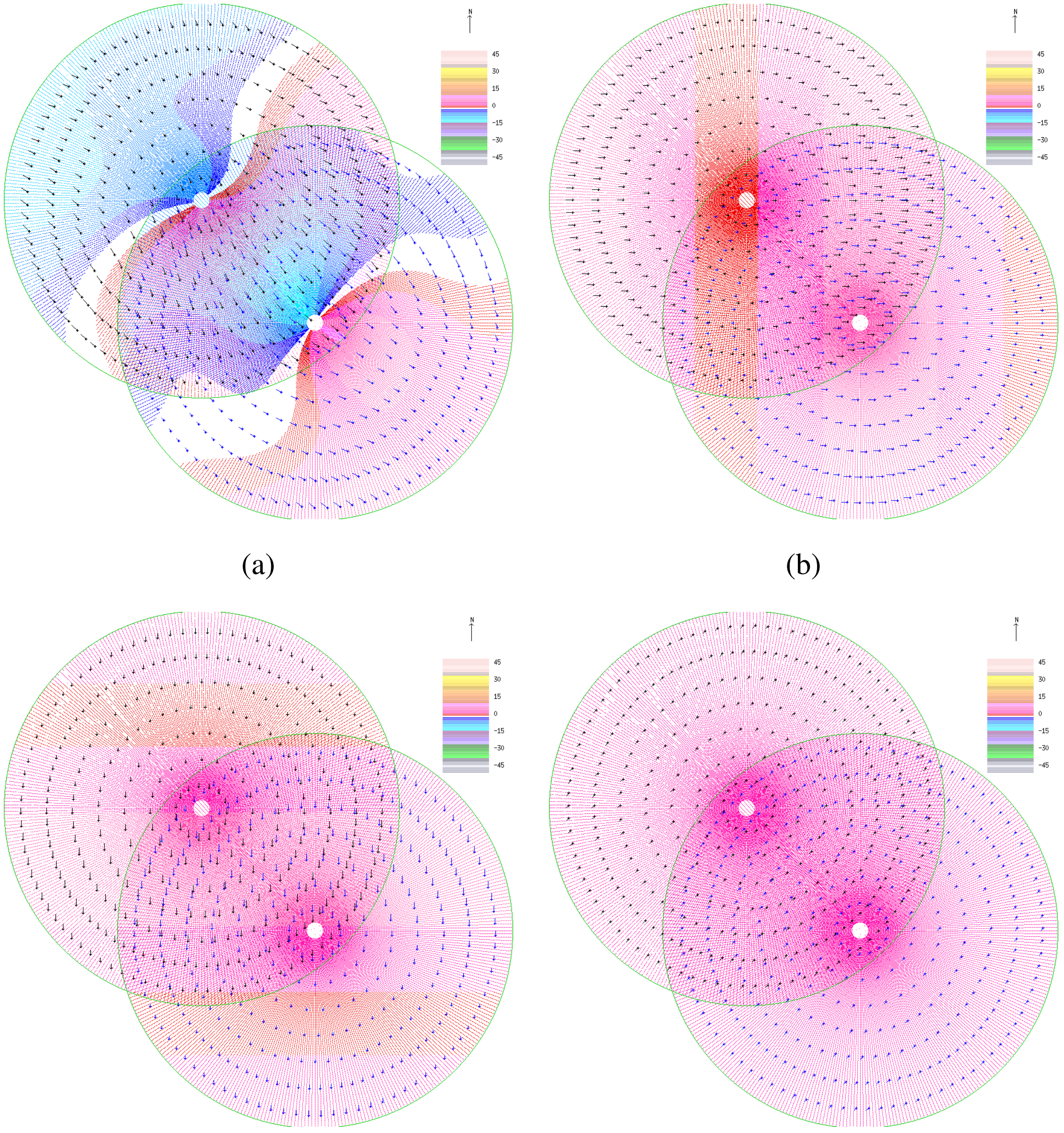
(a)

(b)

(c)

(d)

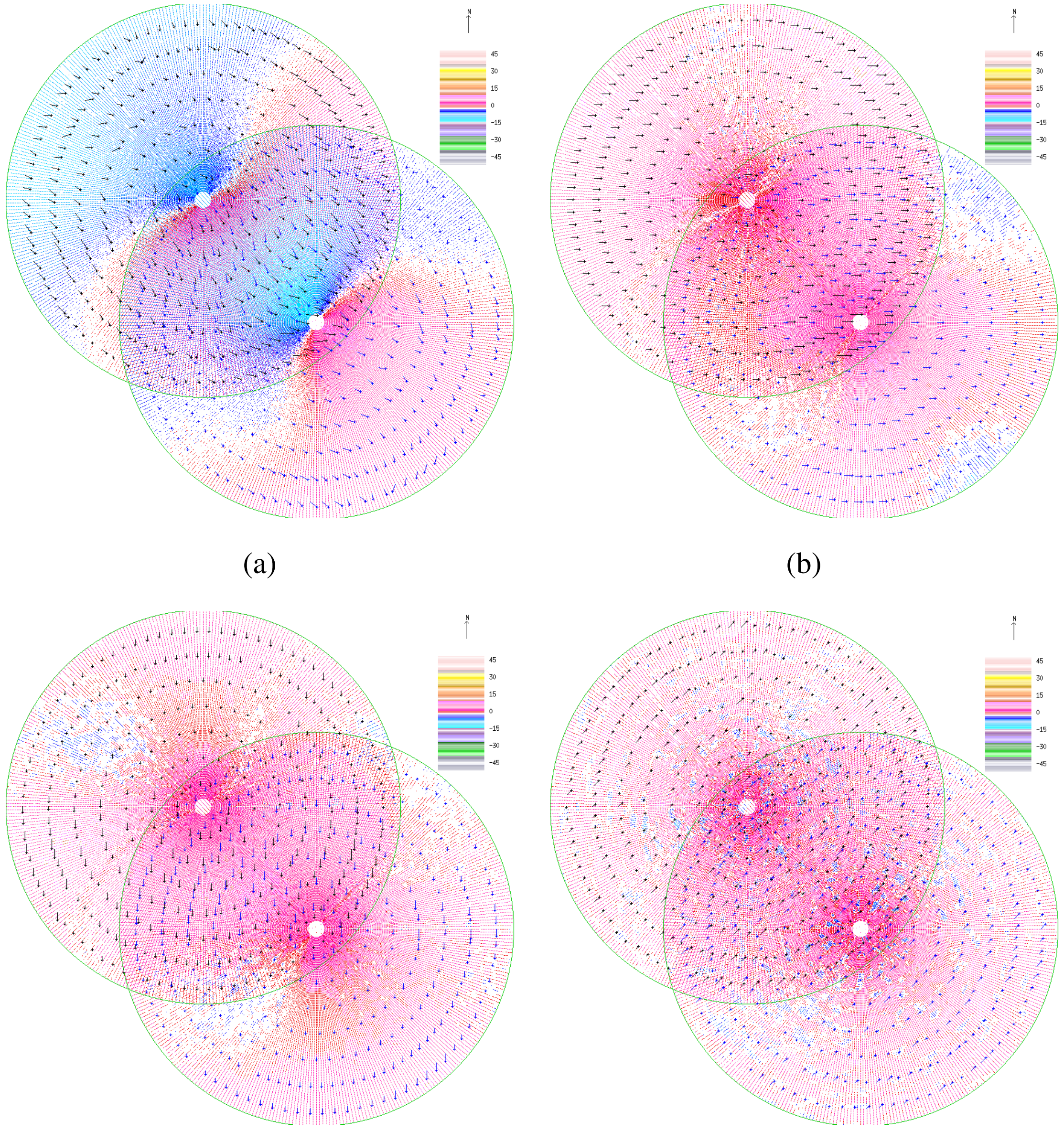
Figure A.17: The dual retrieved synthetic velocity of group 3 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

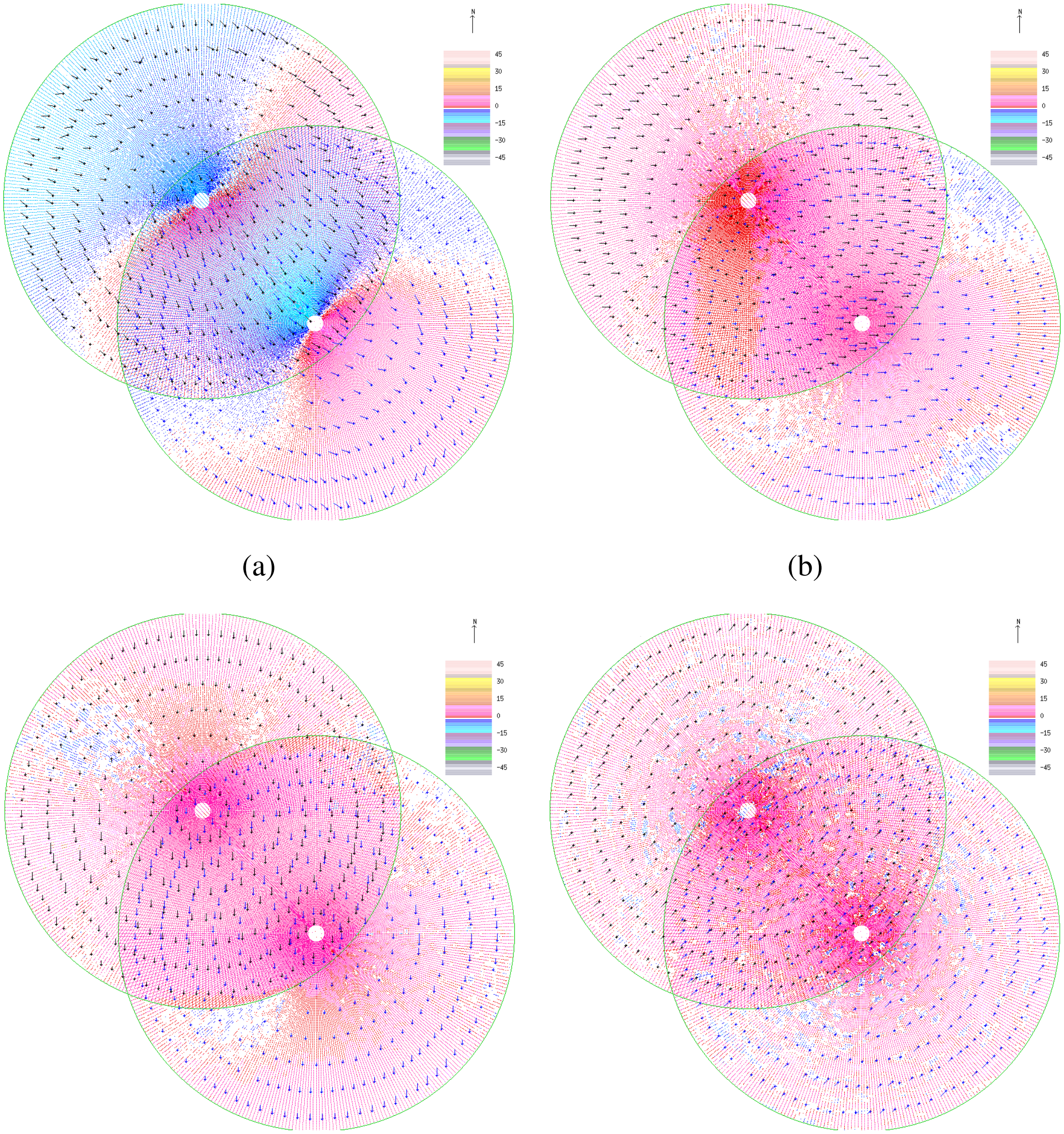
Figure A.18: The correct synthetic velocity of group 3 at variation level $K = 5$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

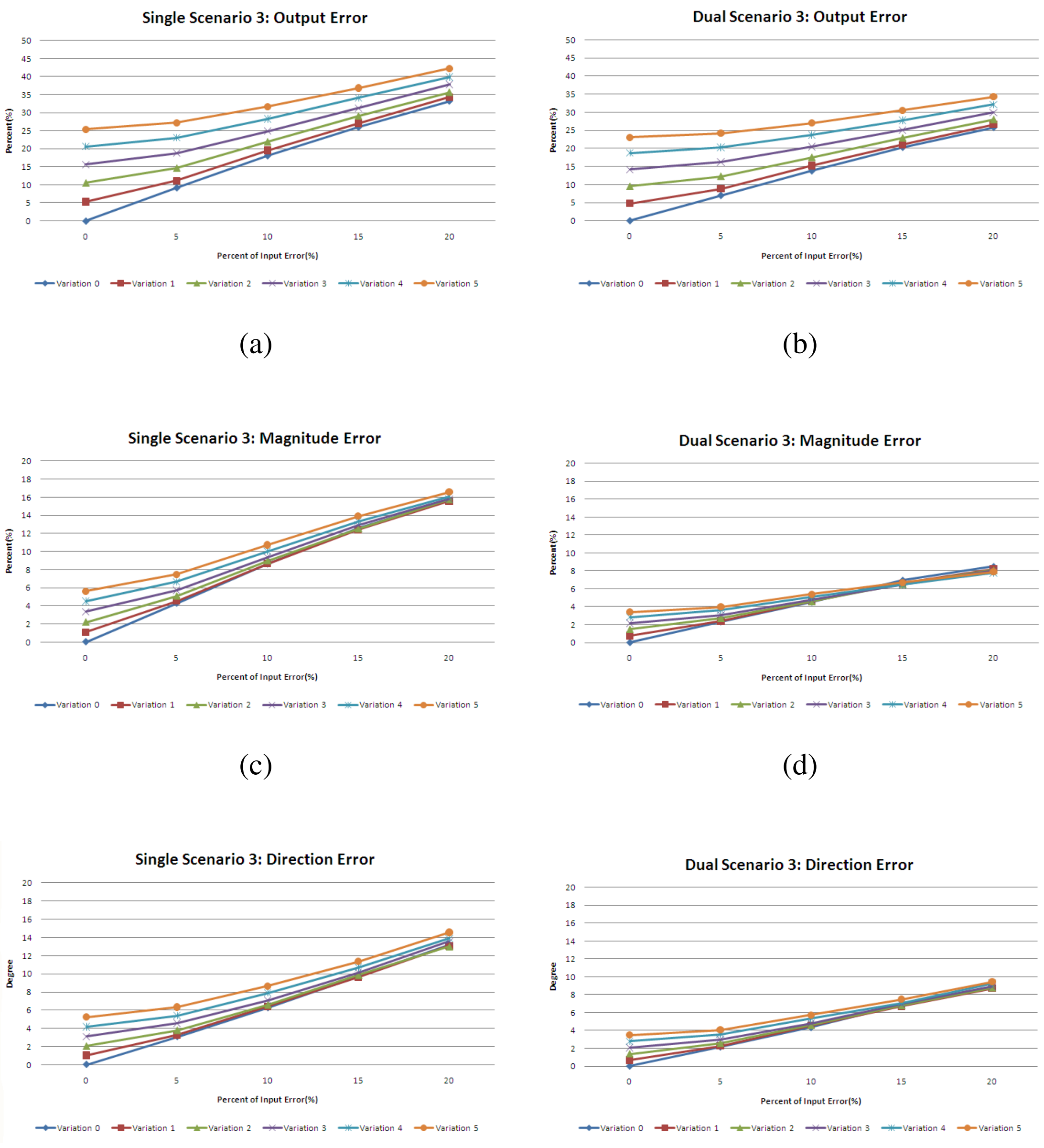
Figure A.19: The single retrieved synthetic velocity of group 3 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

Figure A.20: The dual retrieved synthetic velocity of group 3 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.

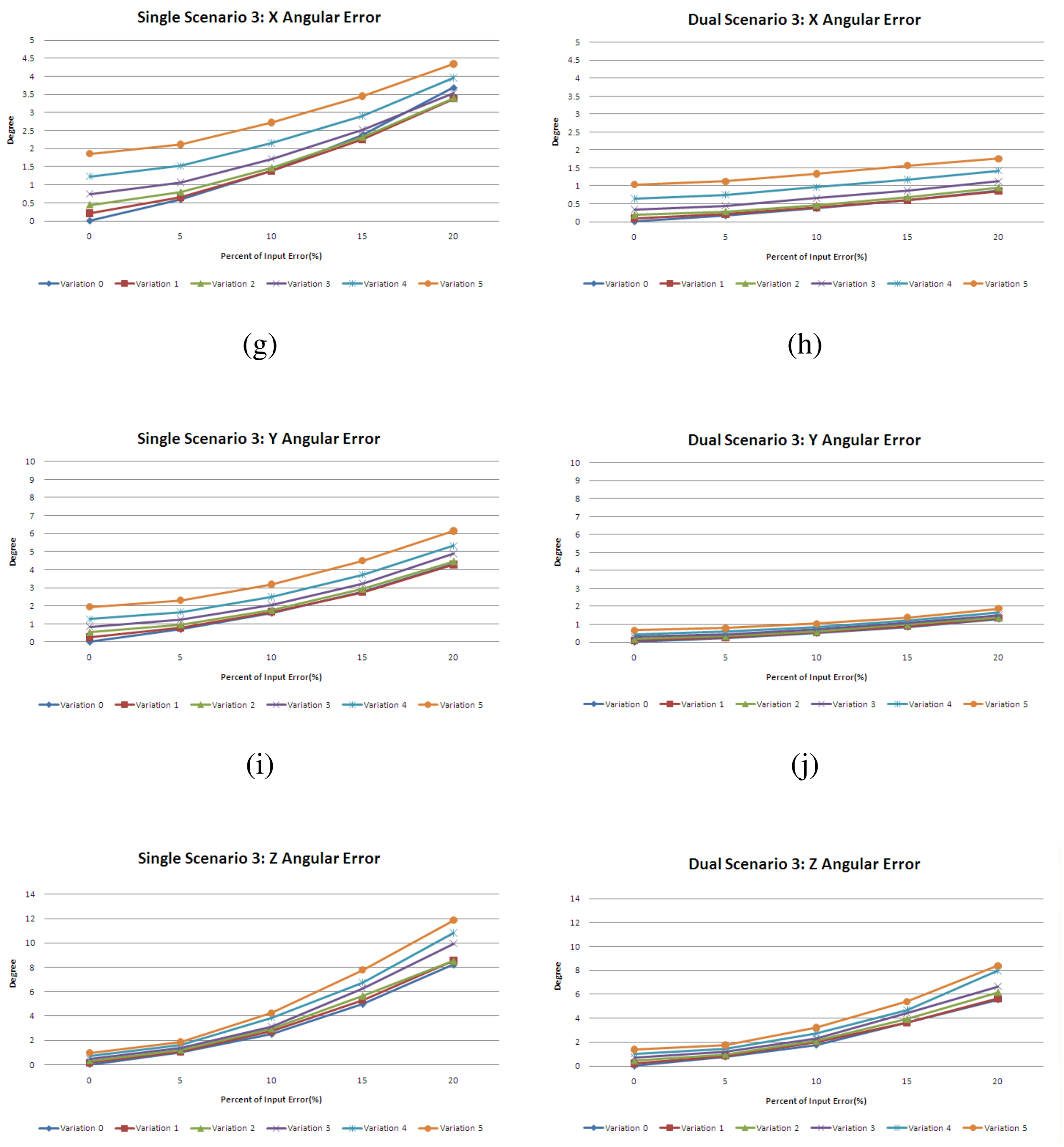


(a) (b)

(c) (d)

(e) (f)

Figure A.21: Result Analysis of Synthetic Data Group 3: (5.0, 5.0, 5.0). The Single Retrieval (a), (c), (e) and Dual Retrieval (b), (d), (f).


Single Scenario 3: X Angular Error

Dual Scenario 3: X Angular Error

Single Scenario 3: Y Angular Error

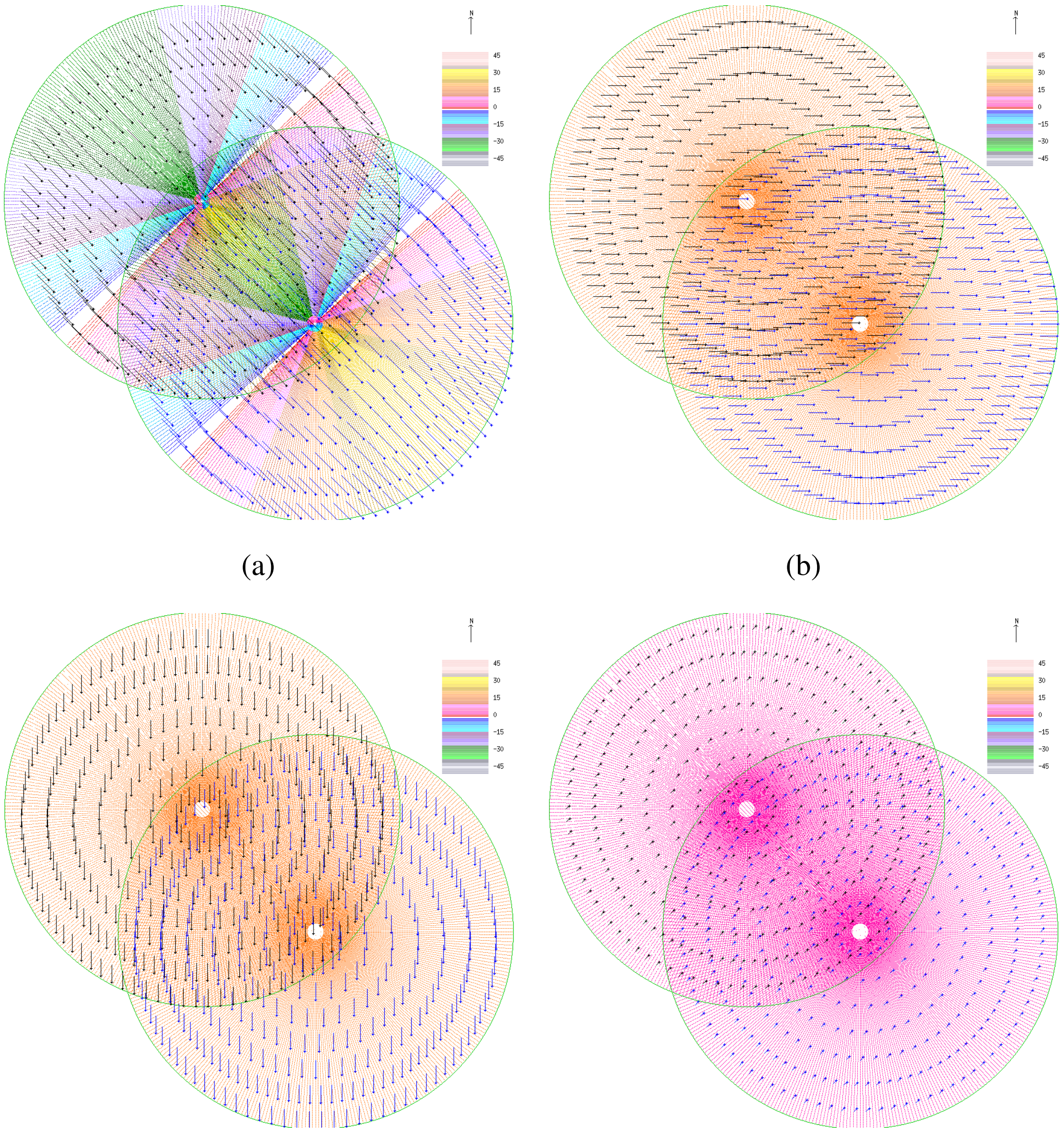
Dual Scenario 3: Y Angular Error

Single Scenario 3: Z Angular Error

Dual Scenario 3: Z Angular Error

(g) (h) (i) (j) (k) (l)

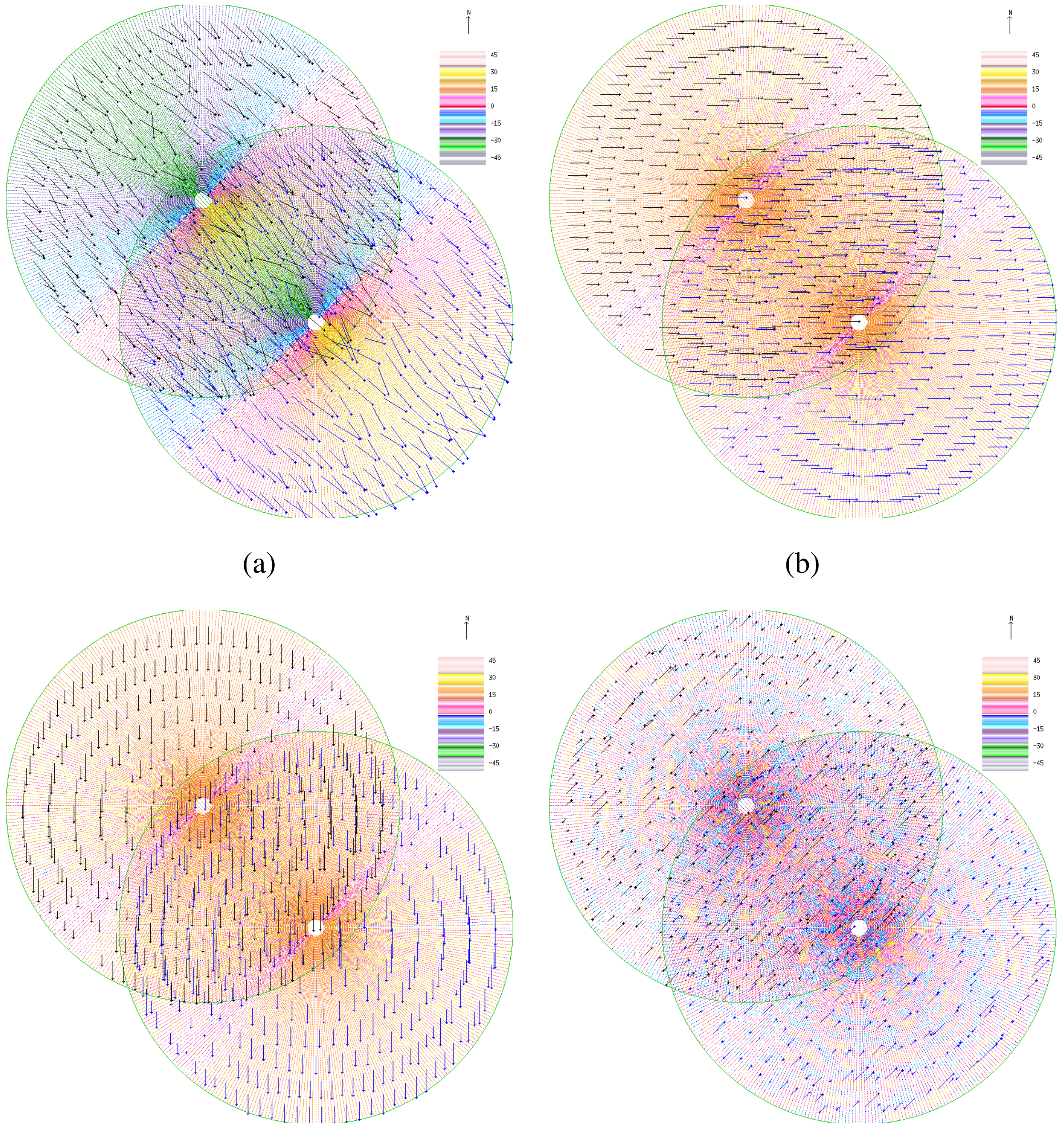
Figure A.21: Result Analysis of Synthetic Data Group 3: (5.0, 5.0, 5.0). The Single Retrieval (g), (i), (k) and Dual retrieval (h), (j), (l).



(a) (b)

(c) (d)

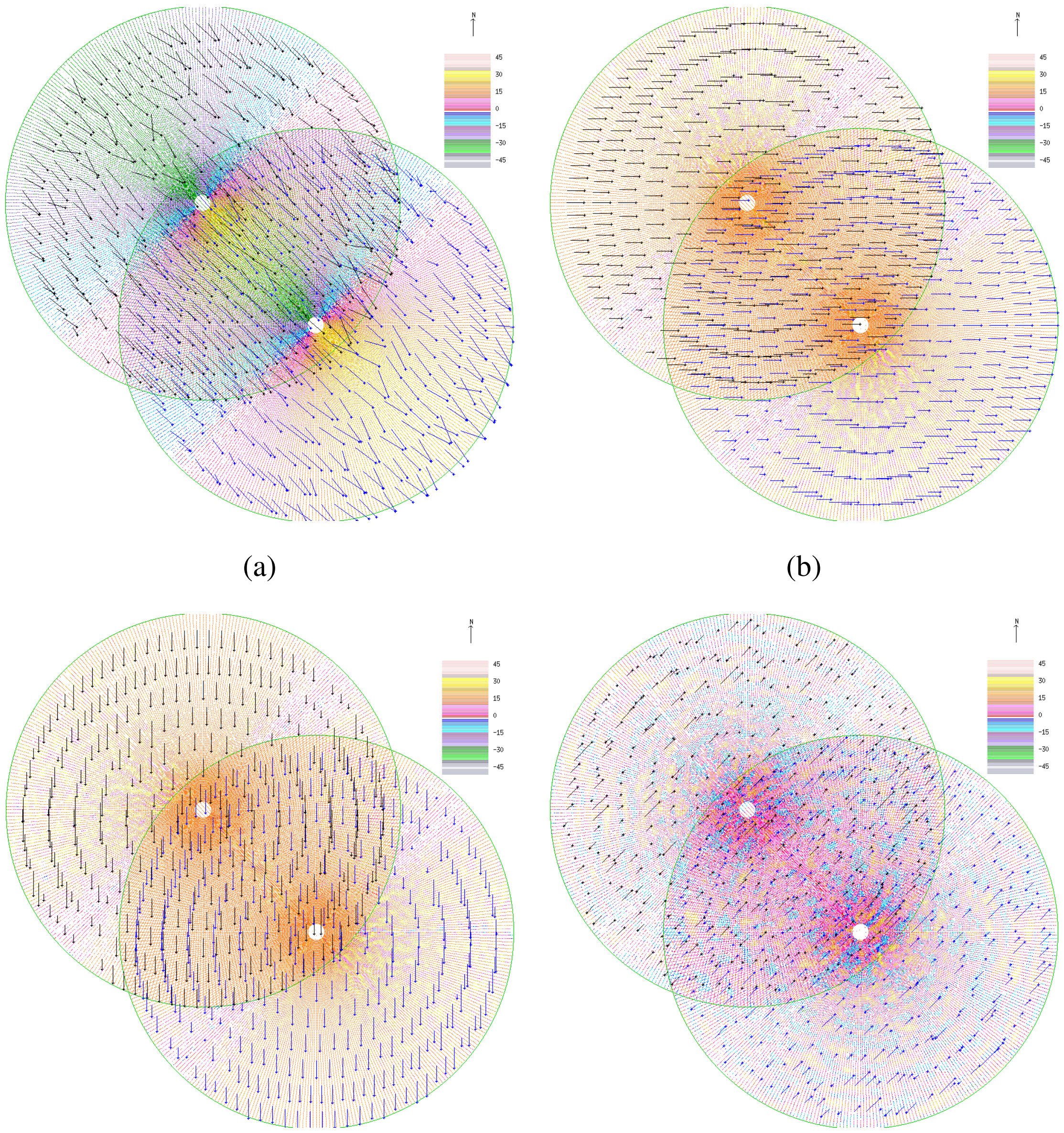
Figure A.22: The correct synthetic velocity of group 4 at variation level $K = 0$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

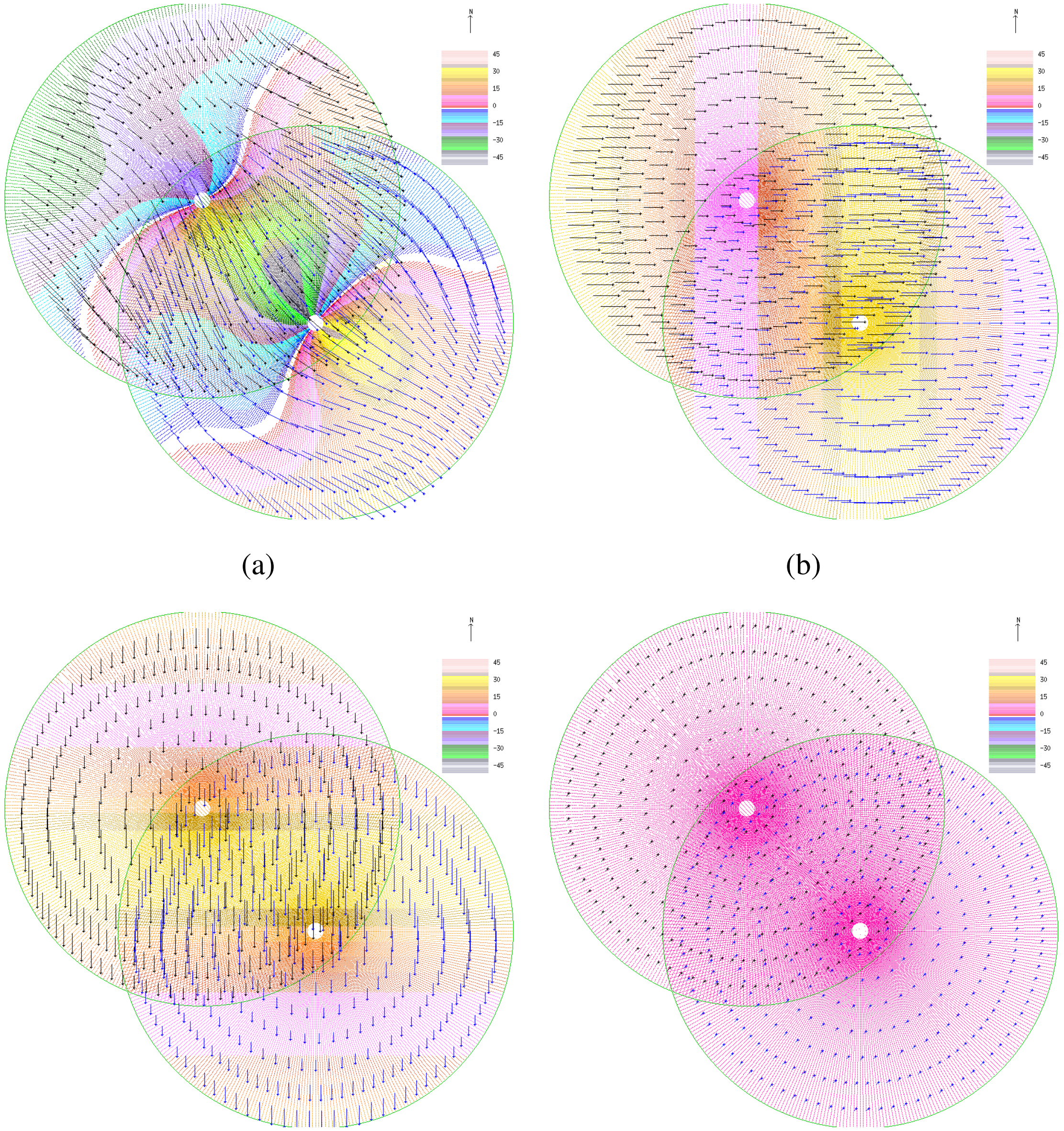
Figure A.23: The single retrieved synthetic velocity of group 4 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

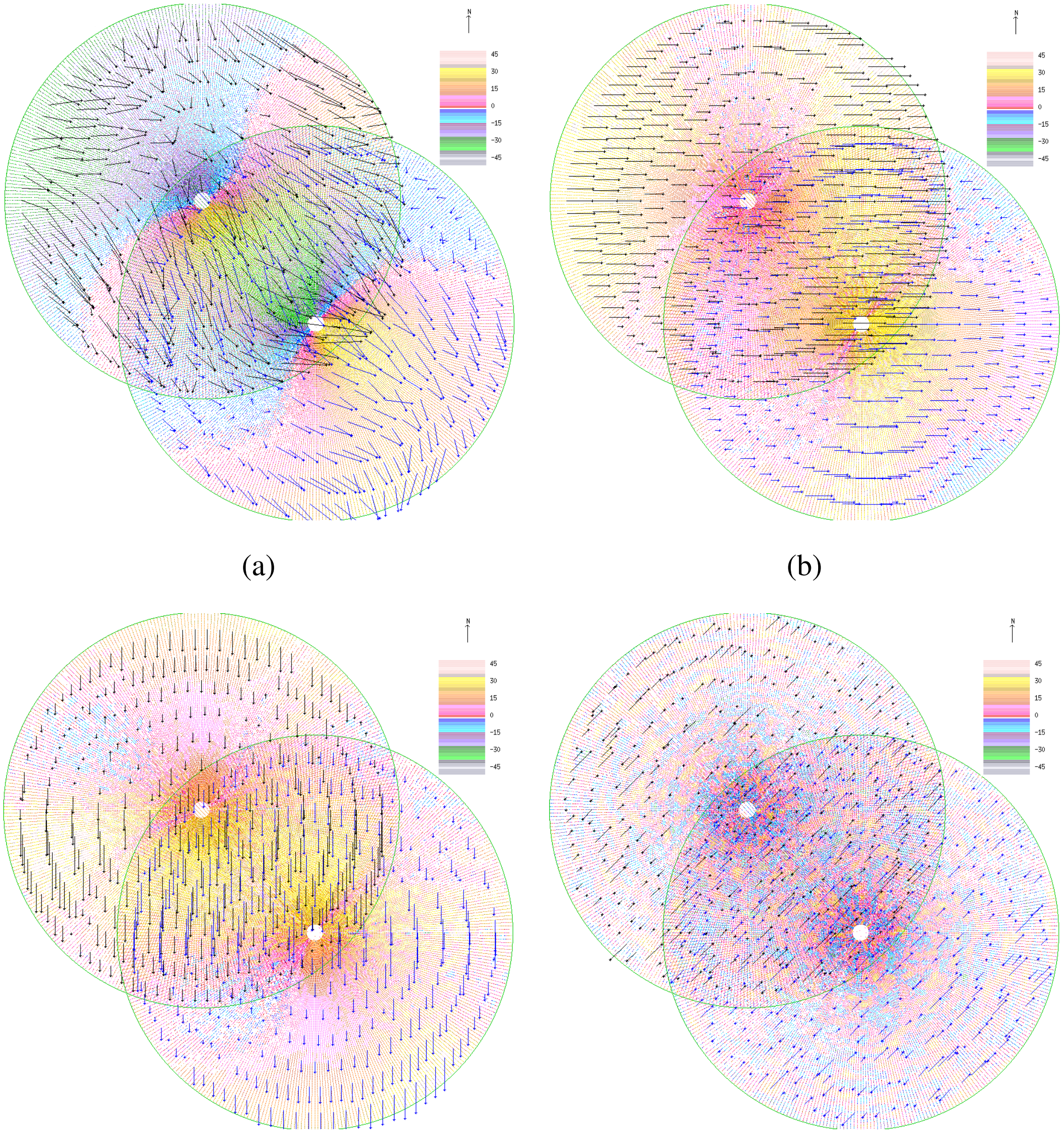
Figure A.24: The dual retrieved synthetic velocity of group 4 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

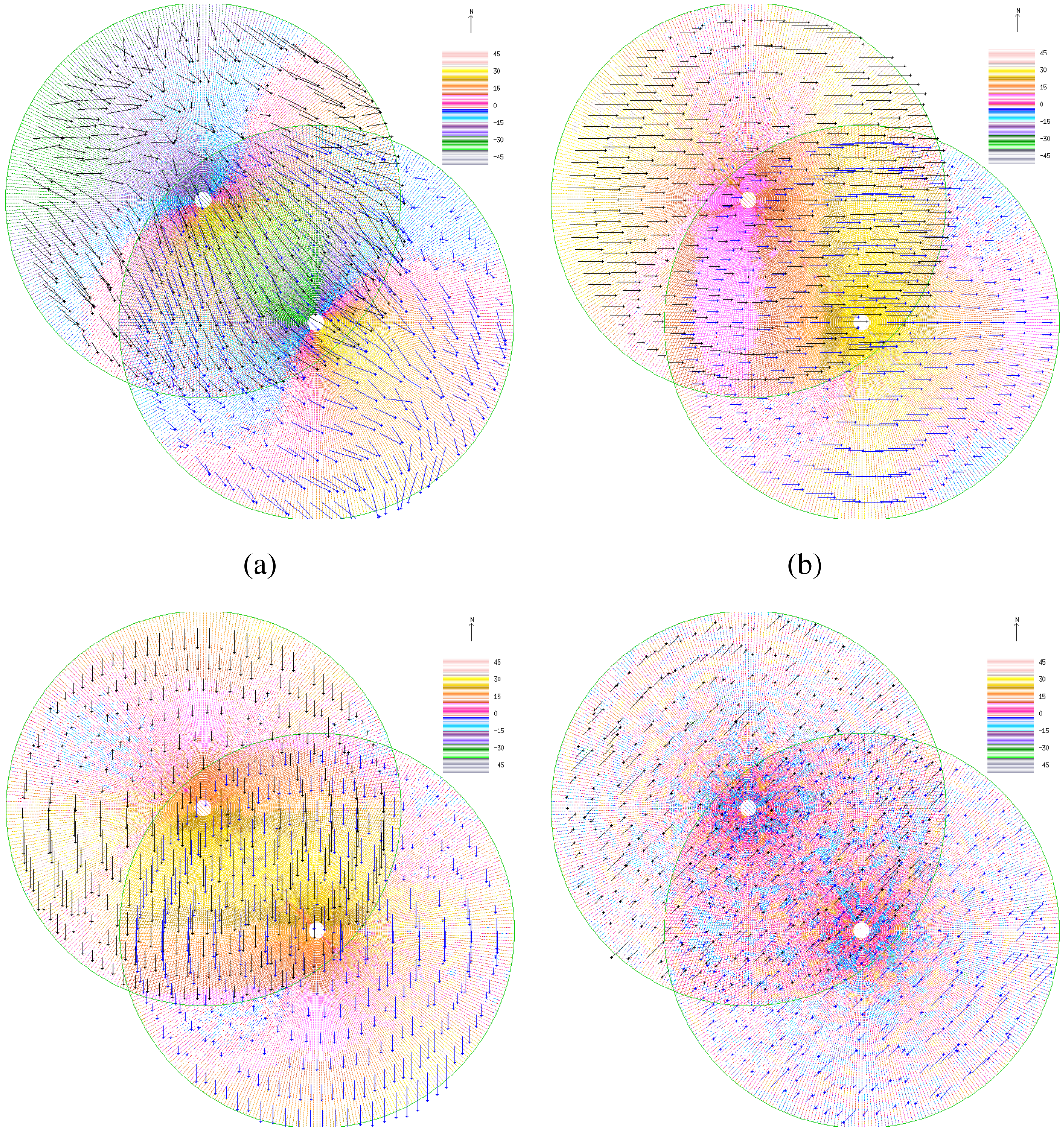
Figure A.25: The correct synthetic velocity of group 4 at variation level $K = 5$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

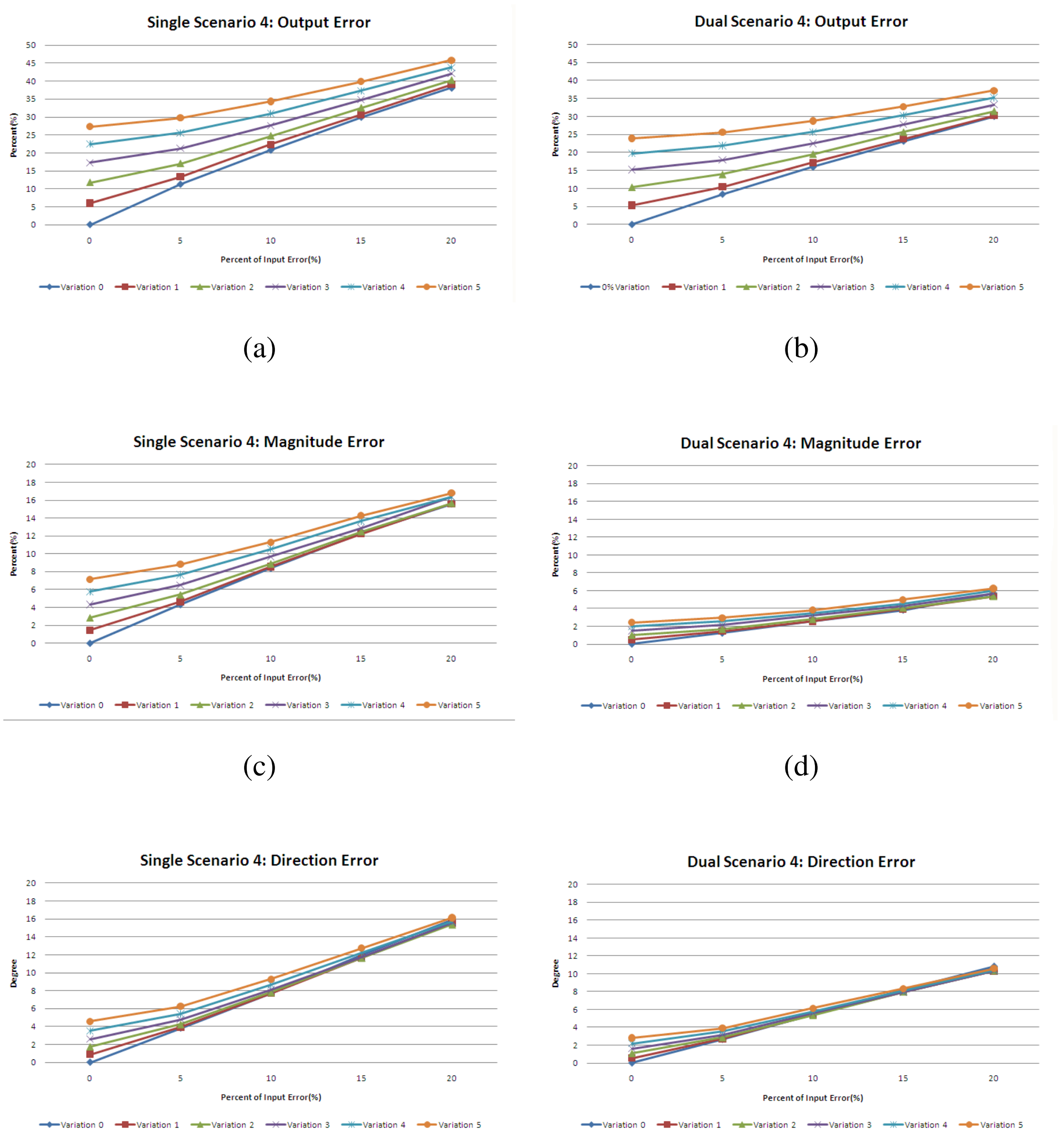
Figure A.26: The single retrieved synthetic velocity of group 4 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

Figure A.27: The dual retrieved synthetic velocity of group 4 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.

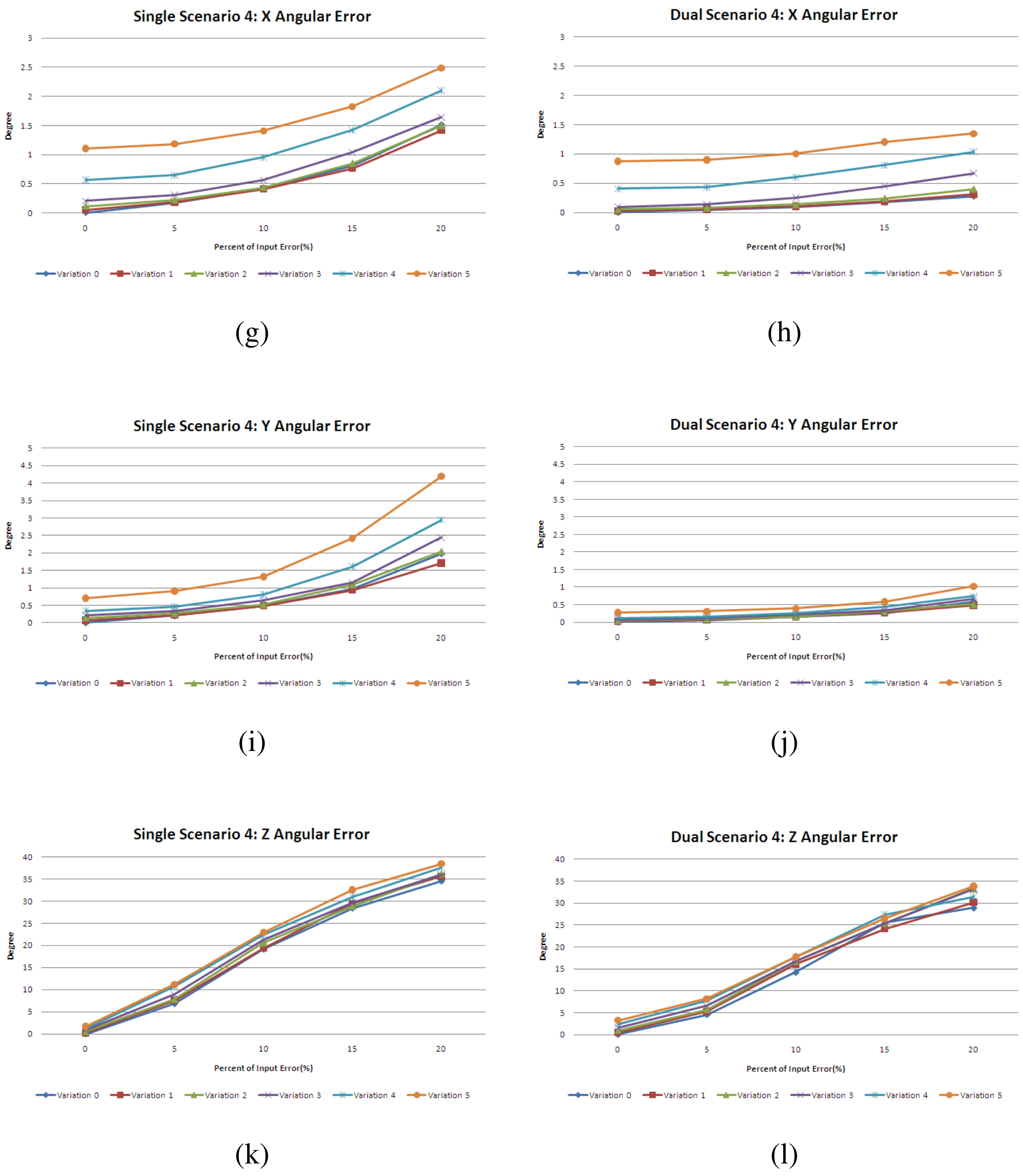


(a) (b)

(c) (d)

(e) (f)

Figure A.28: Result Analysis of Synthetic Data Group 4: (20.0, 20.0, 5.0). The Single Retrieval (a), (c), (e) and Dual Retrieval (b), (d), (f).


Single Scenario 4: X Angular Error

Dual Scenario 4: X Angular Error

(g)

(h)

Single Scenario 4: Y Angular Error

Dual Scenario 4: Y Angular Error

(i)

(j)

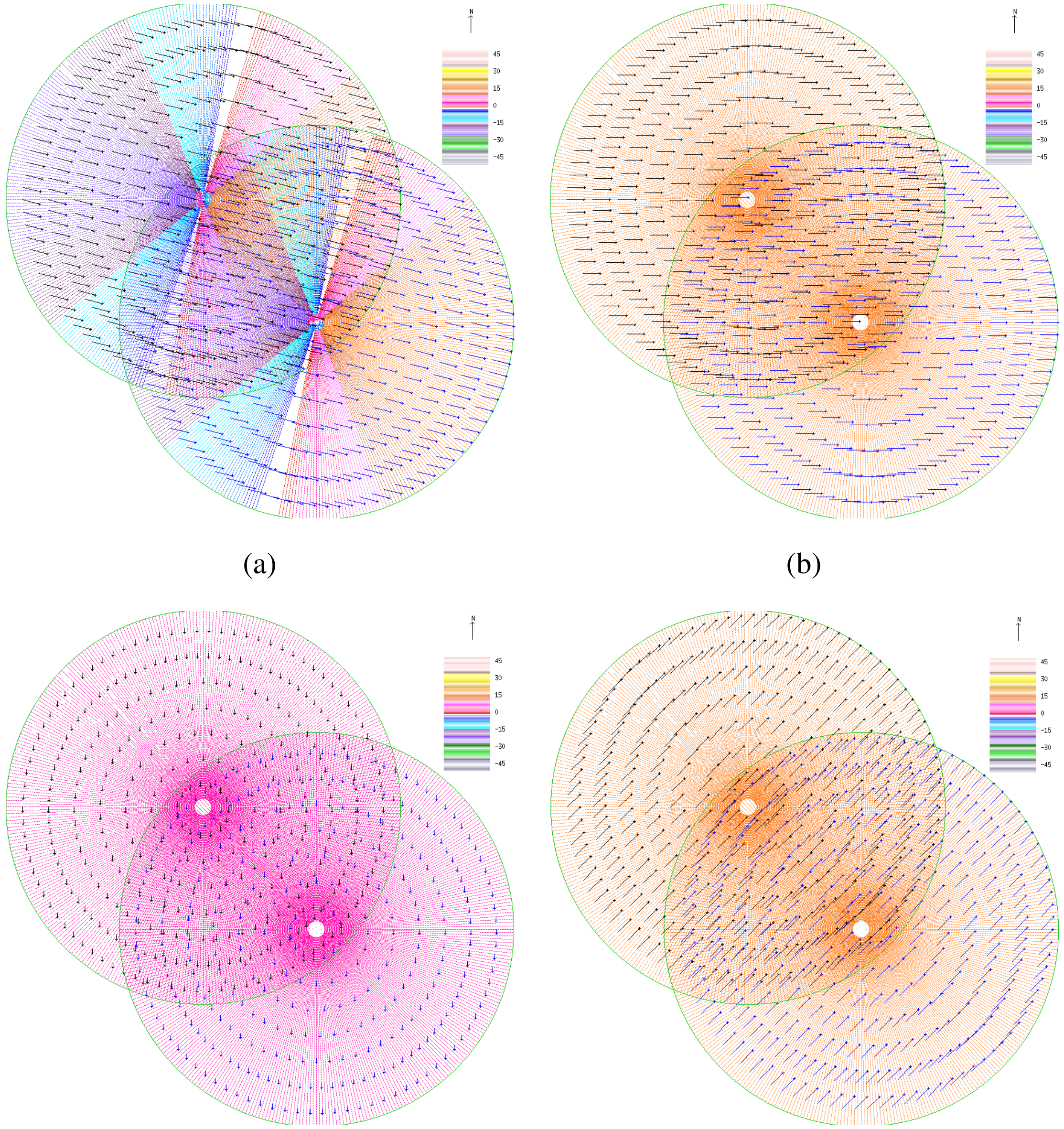
Single Scenario 4: Z Angular Error

Dual Scenario 4: Z Angular Error

(k)

(l)

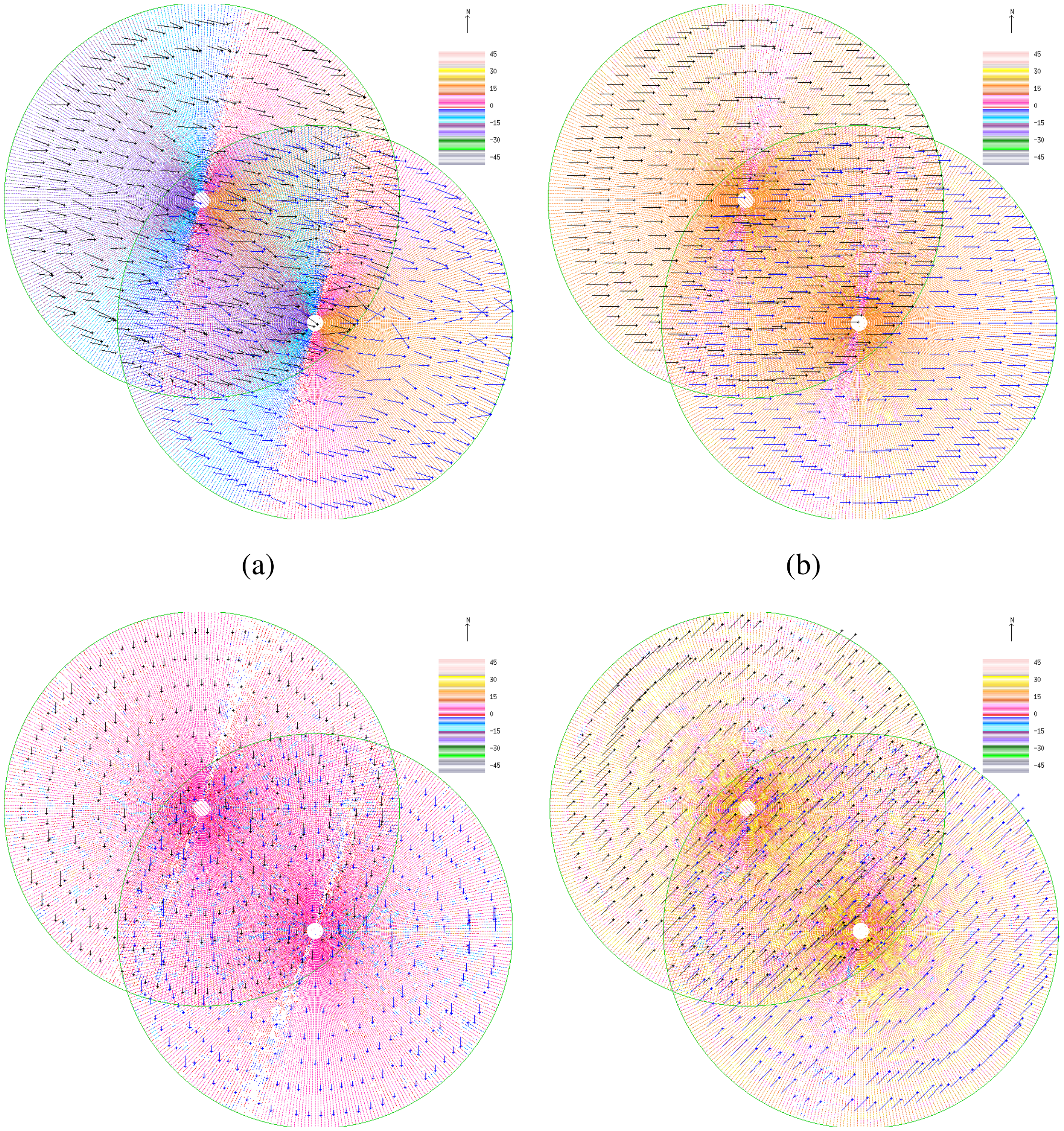
Figure A.28: Result Analysis of Synthetic Data Group 4: (20.0, 20.0, 5.0). The Single Retrieval (g), (i), (k) and Dual retrieval (h), (j), (l).



(a) (b)

(c) (d)

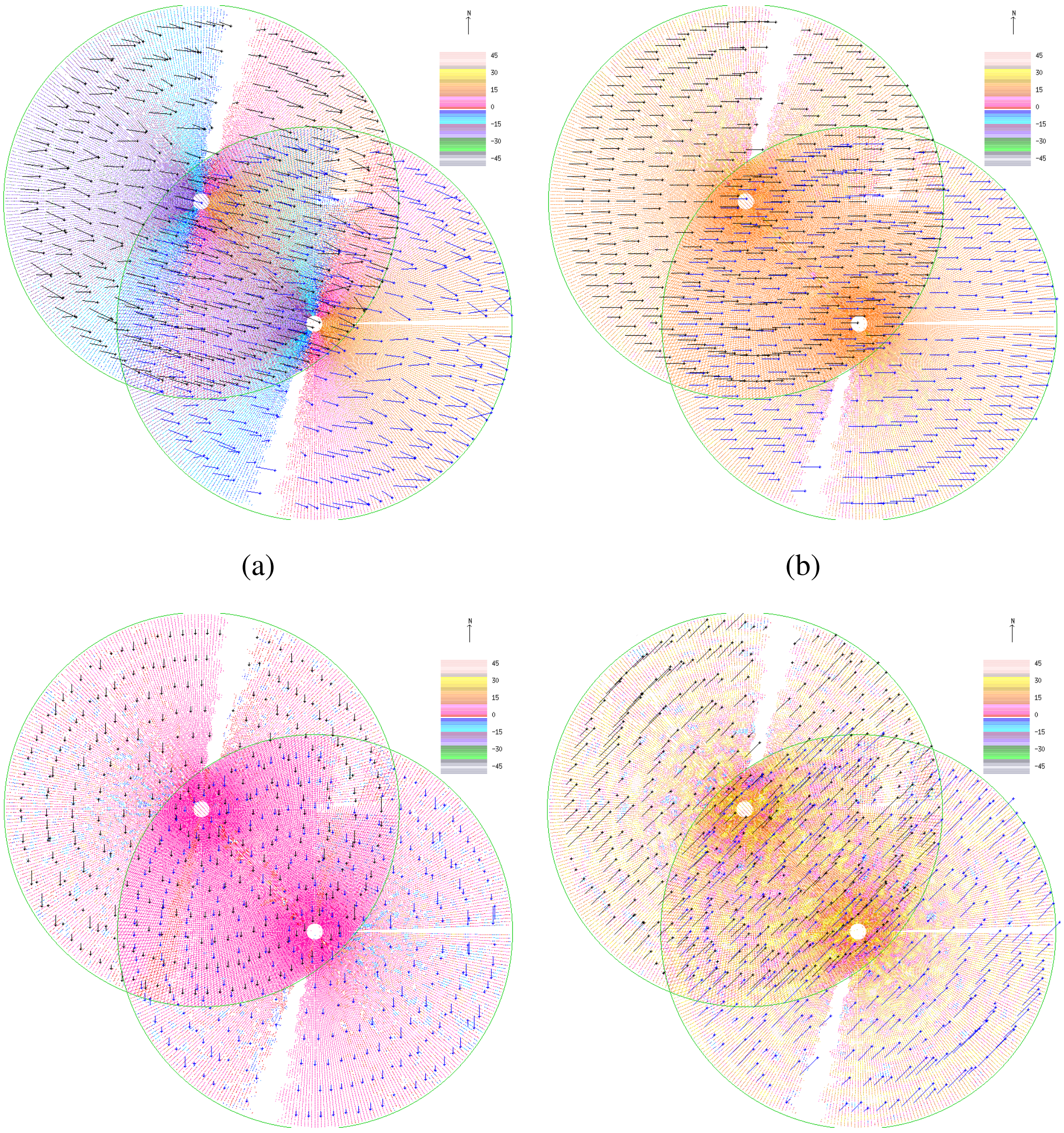
Figure A.29: The correct synthetic velocity of group 5 at variation level $K = 0$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

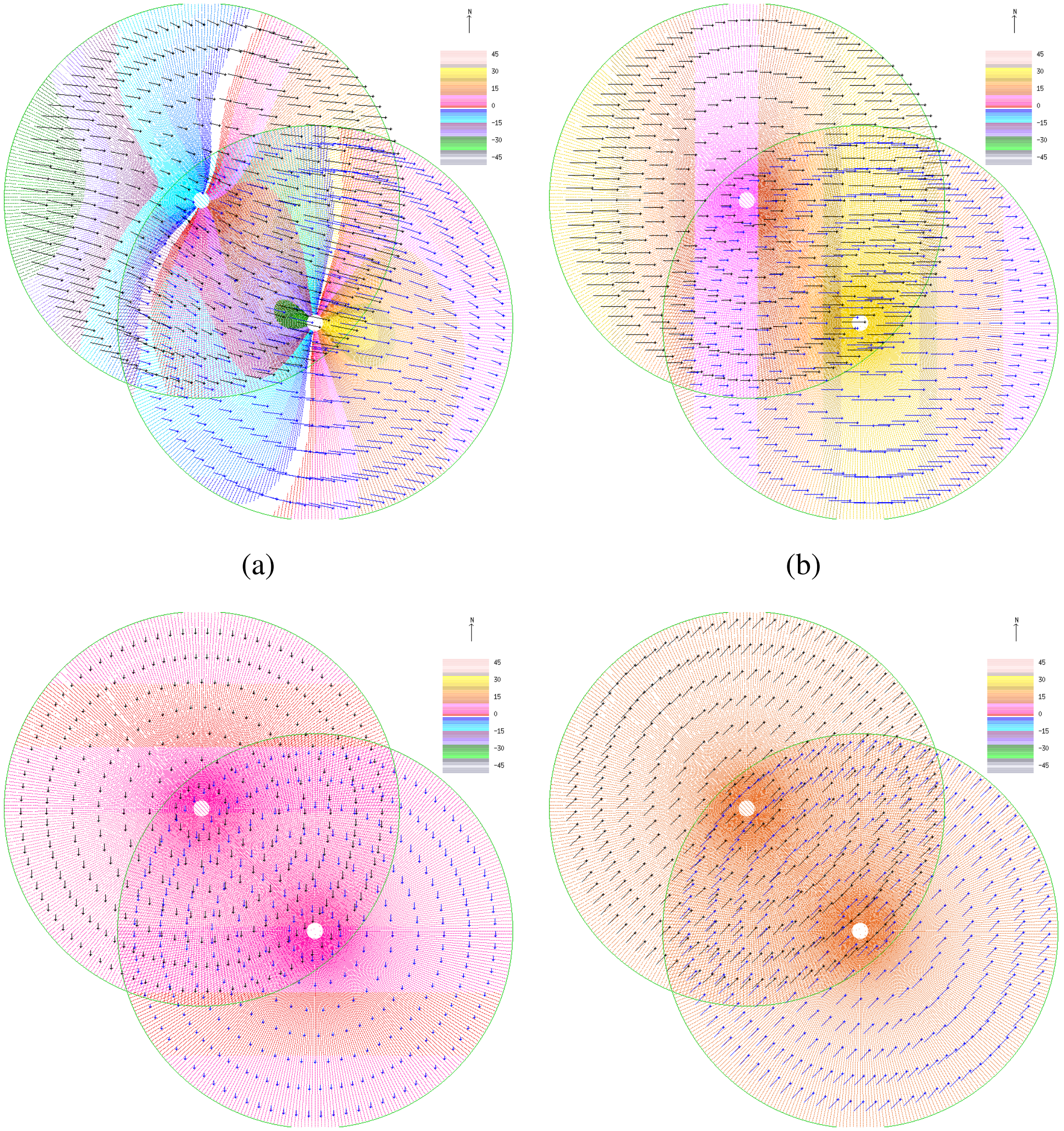
Figure A.30: The single retrieved synthetic velocity of group 5 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

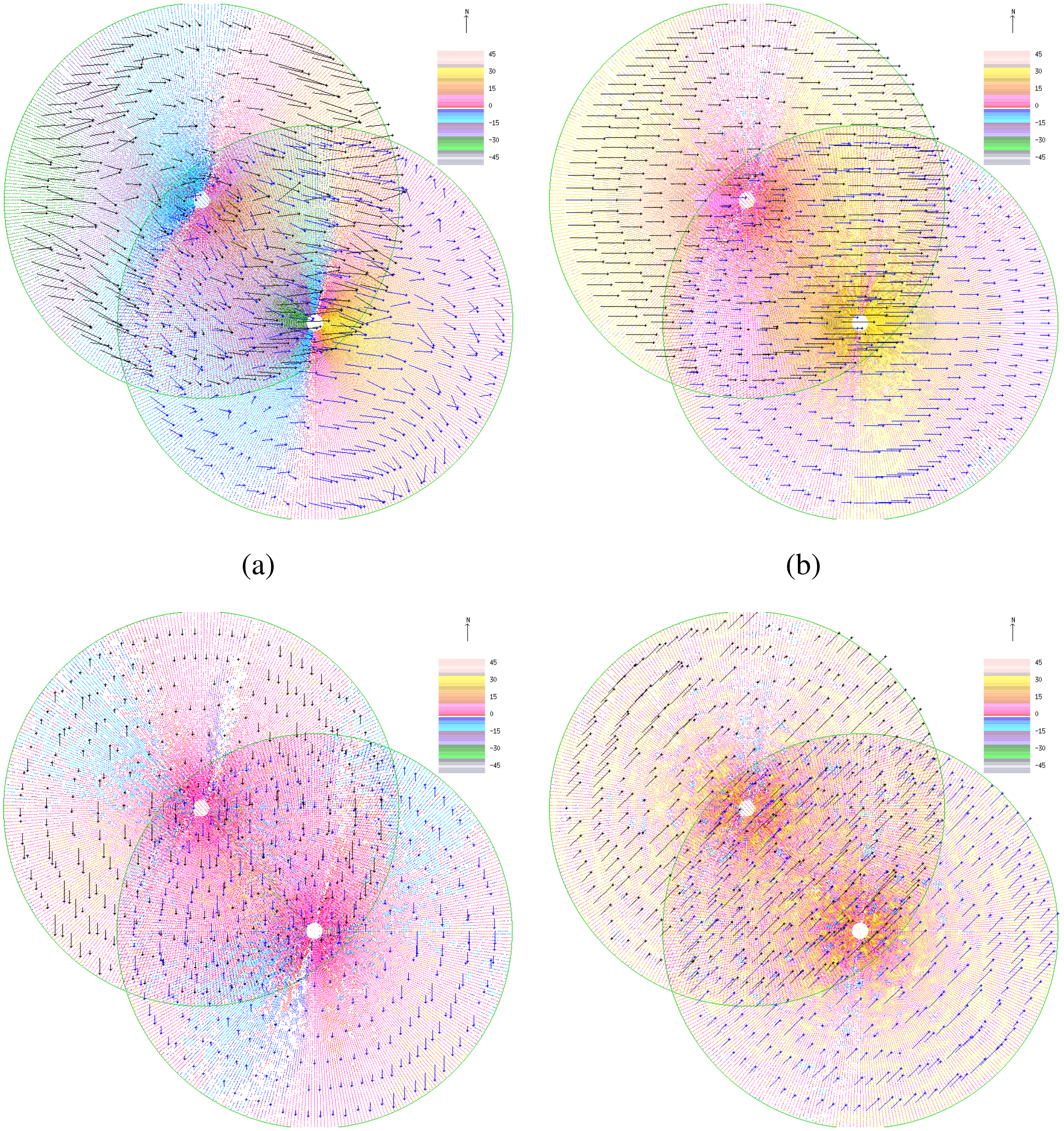
Figure A.31: The dual retrieved synthetic velocity of group 5 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

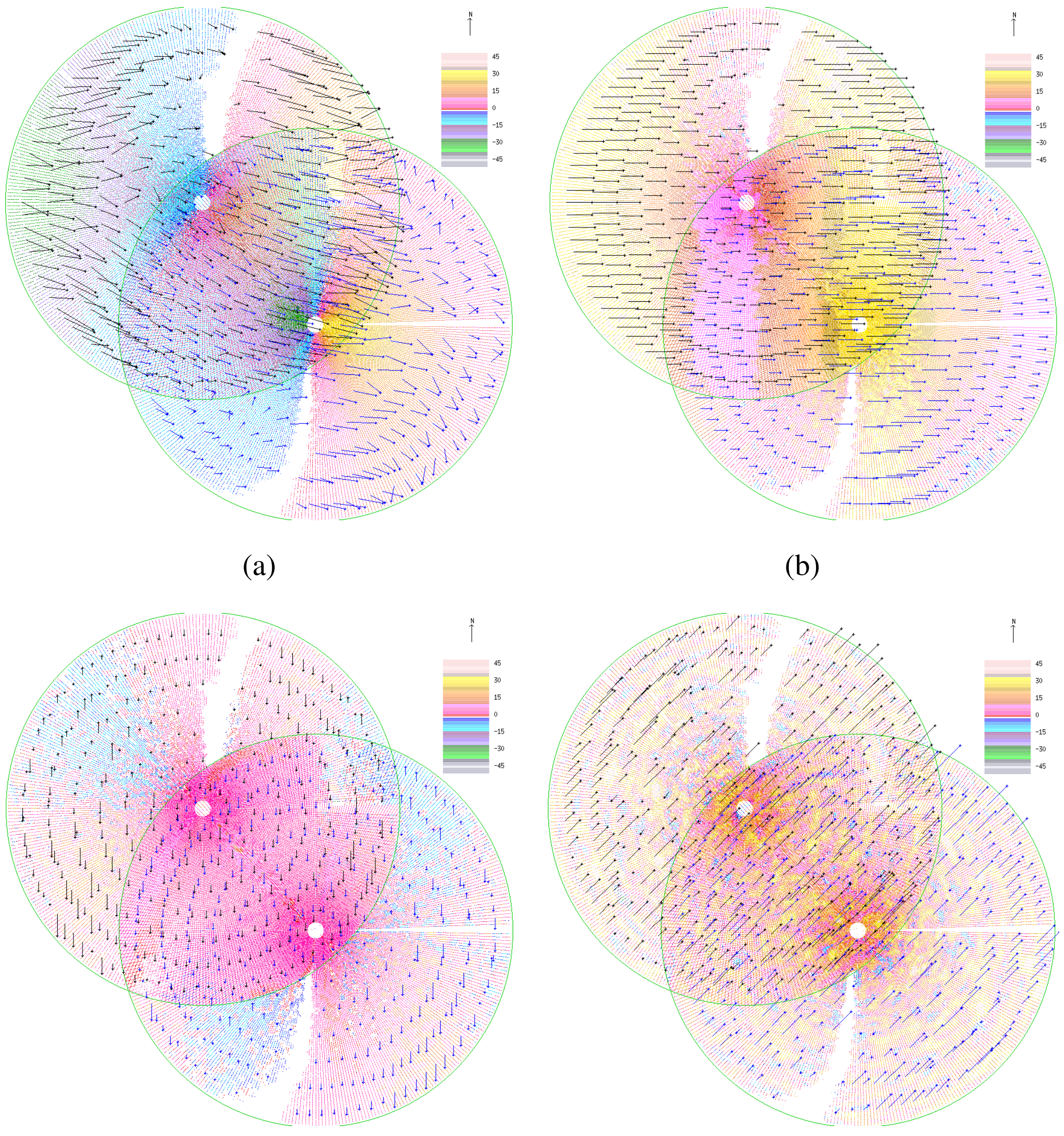
Figure A.32: The correct synthetic velocity of group 5 at variation level $K = 5$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

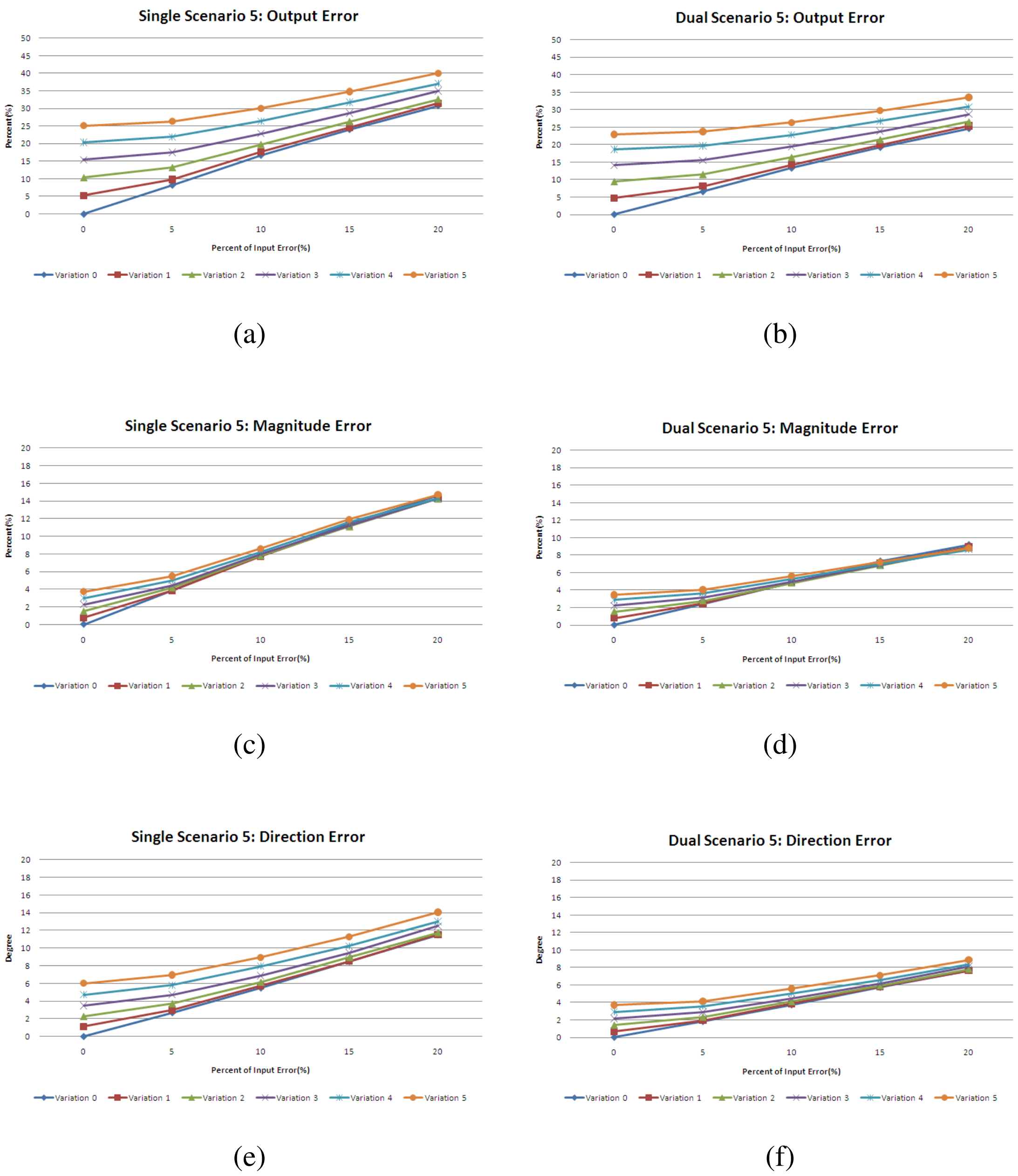
Figure A.33: The single retrieved synthetic velocity of group 5 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

Figure A.34: The dual retrieved synthetic velocity of group 5 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.

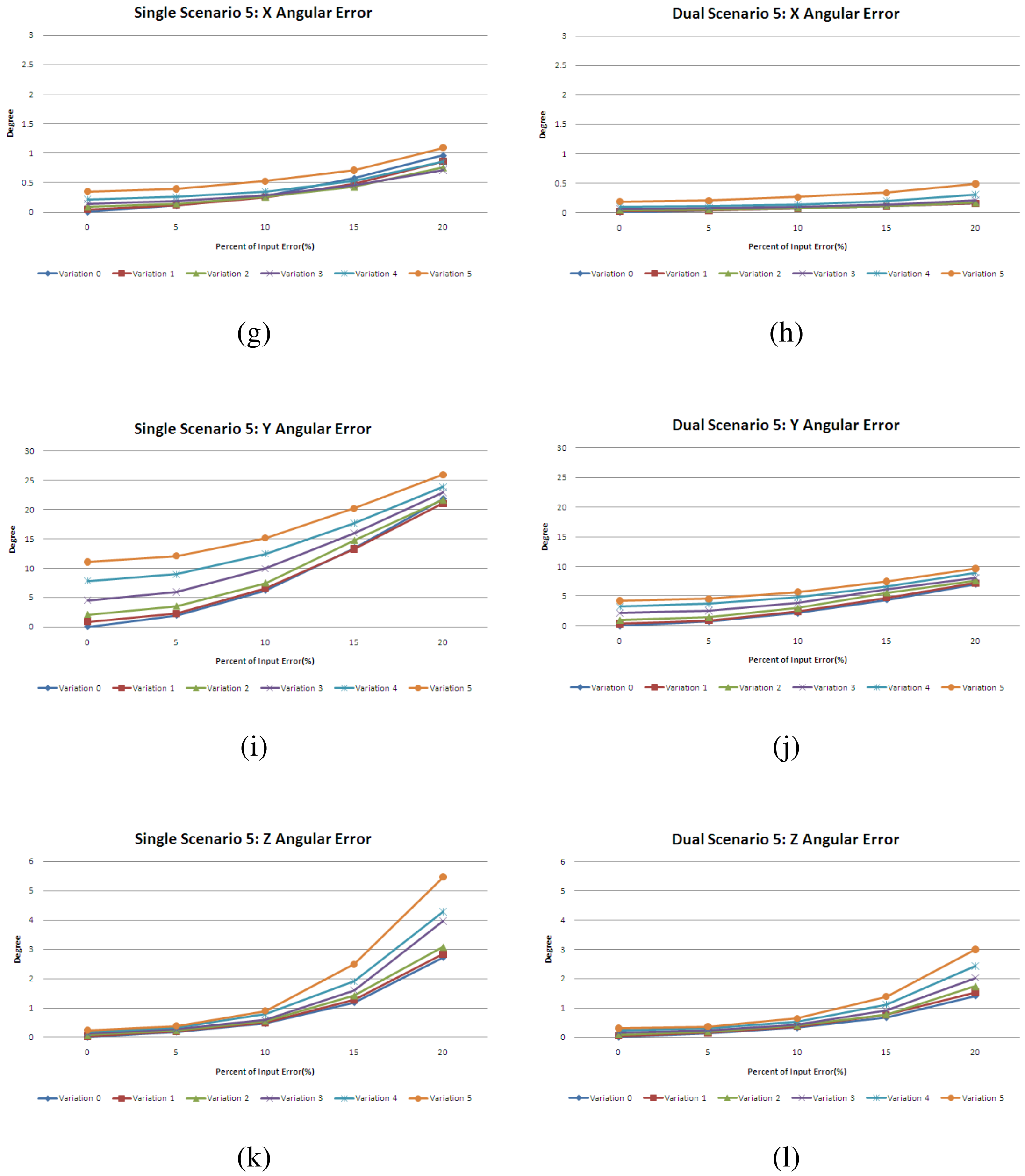


(a) (b)

(c) (d)

(e) (f)

Figure A.35: Result Analysis of Synthetic Data Group 5: (20.0, 5.0, 20.0). The Single Retrieval (a), (c), (e) and Dual Retrieval (b), (d), (f).

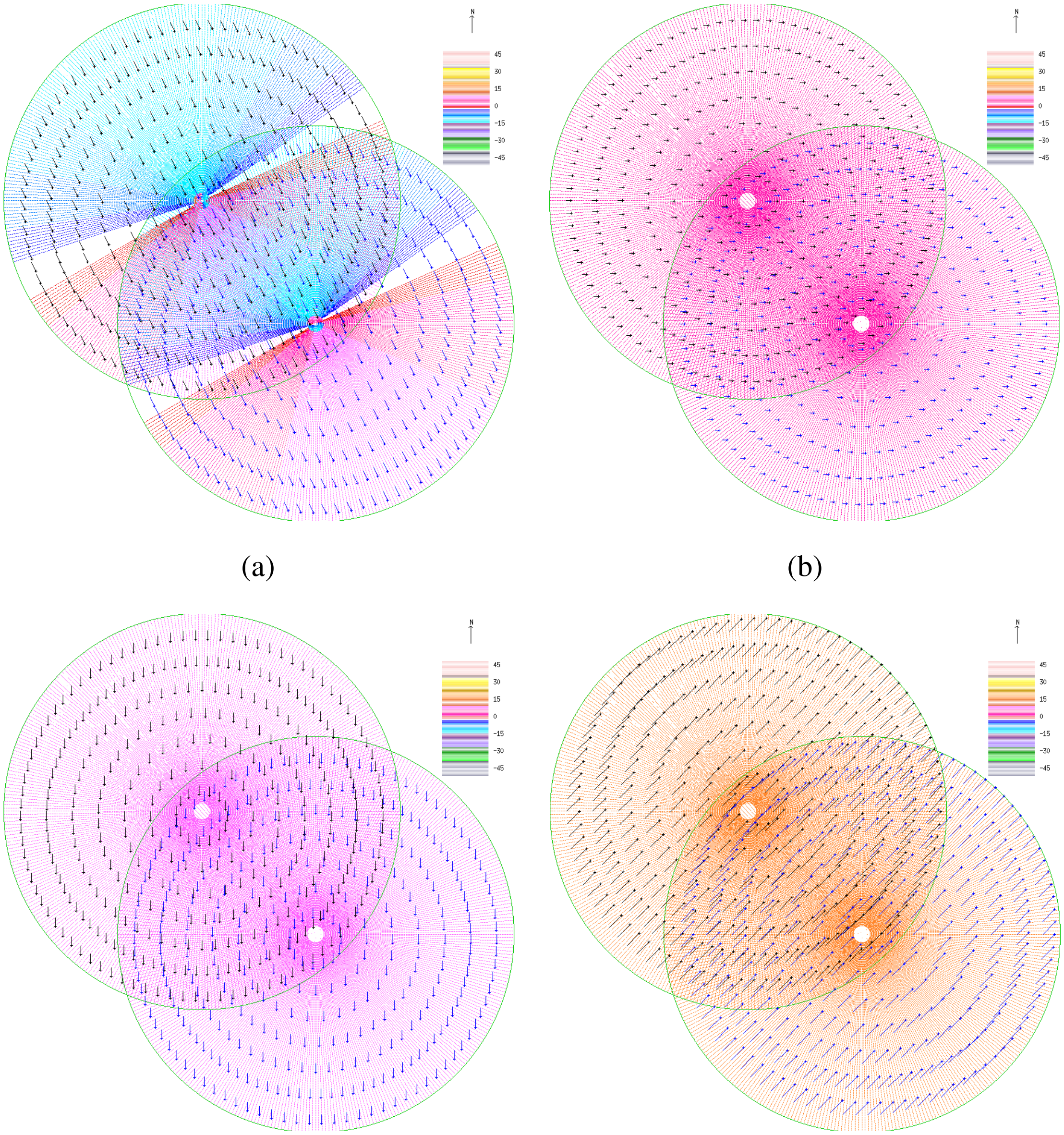


(g) (h)

(i) (j)

(k) (l)

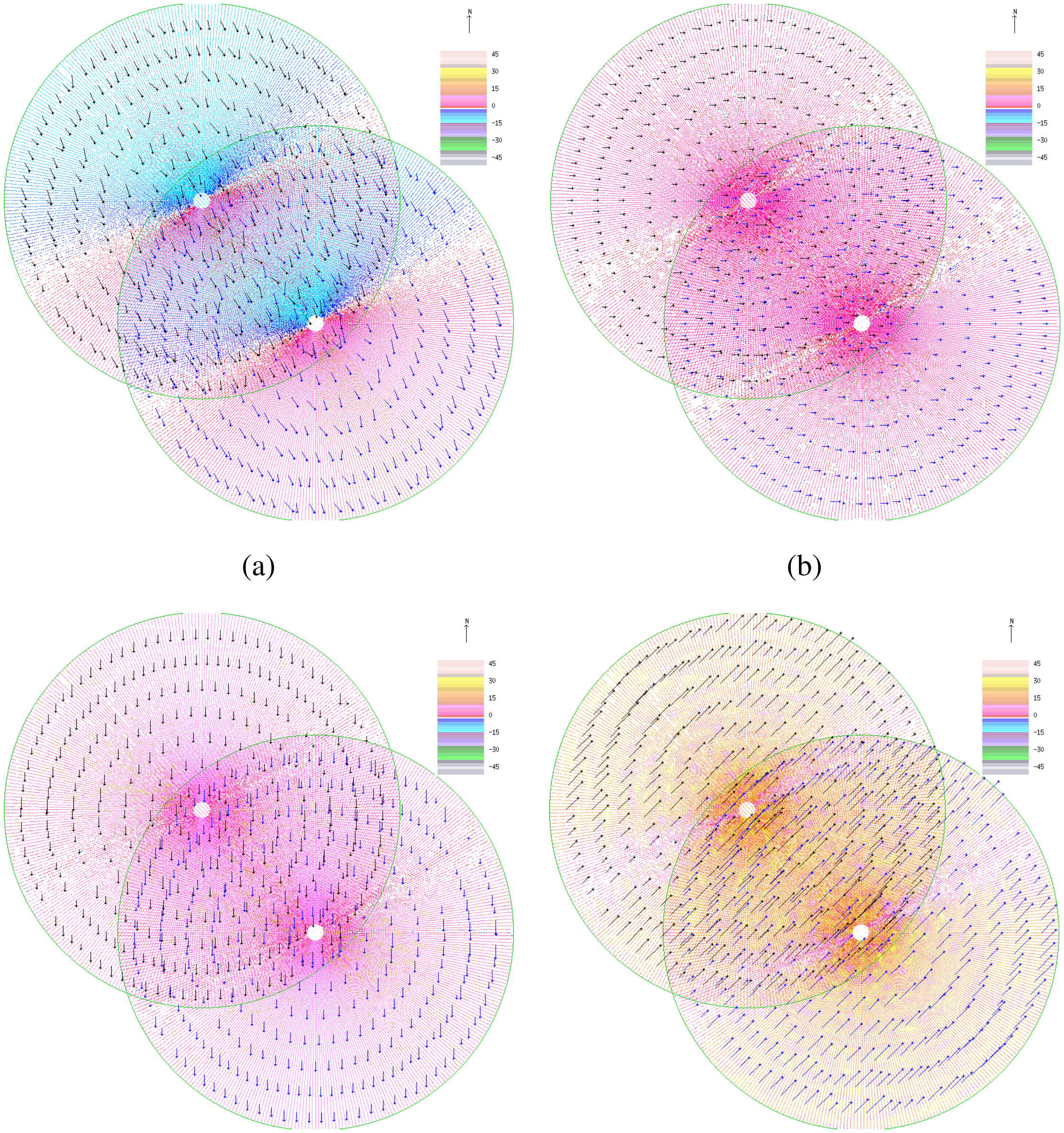
Figure A.35: Result Analysis of Synthetic Data Group 5: (20.0, 5.0, 20.0). The Single Retrieval (g), (i), (k) and Dual retrieval (h), (j), (l).



(a) (b)

(c) (d)

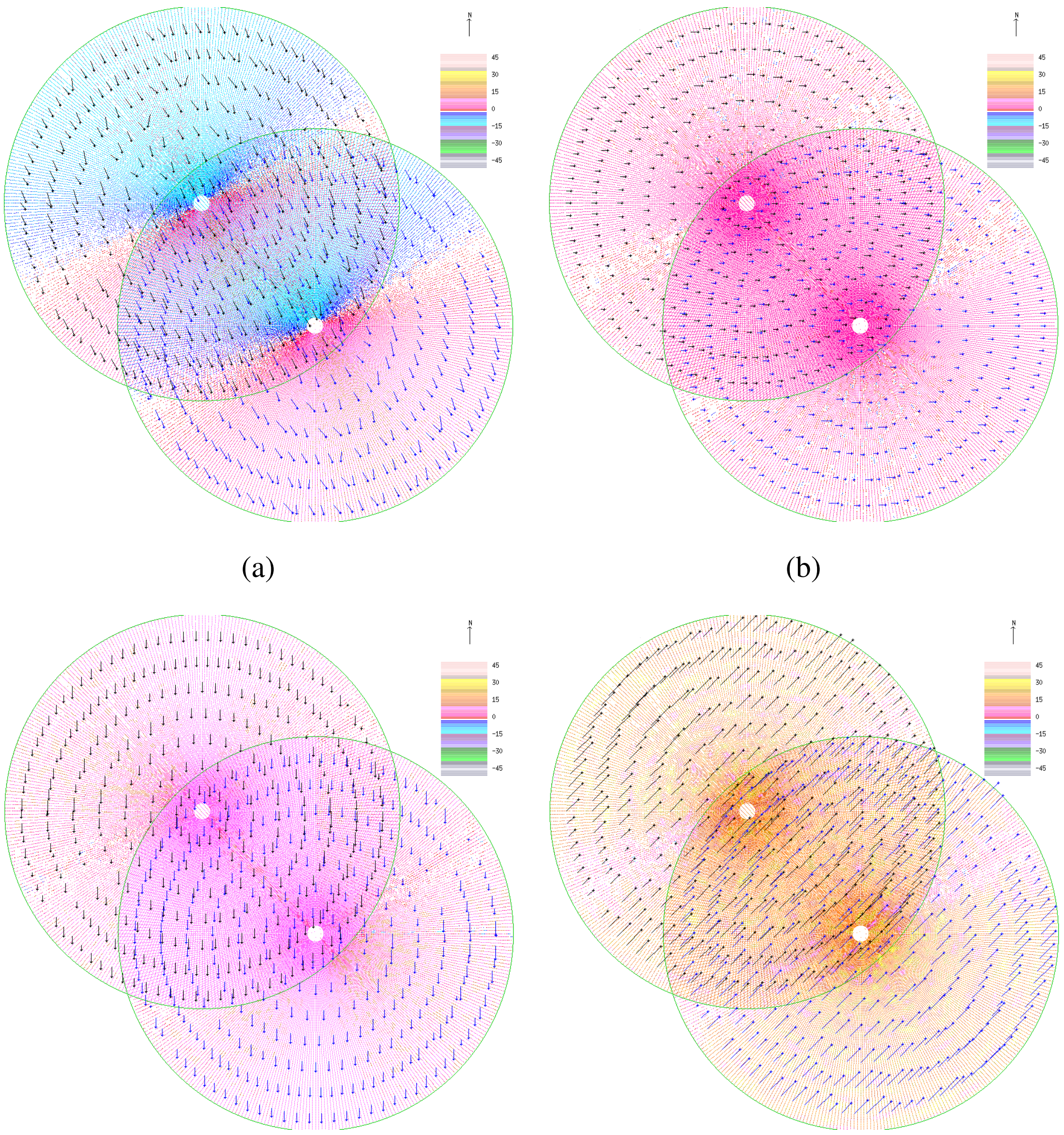
Figure A.36: The correct synthetic velocity of group 2 at variation level $K = 0$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

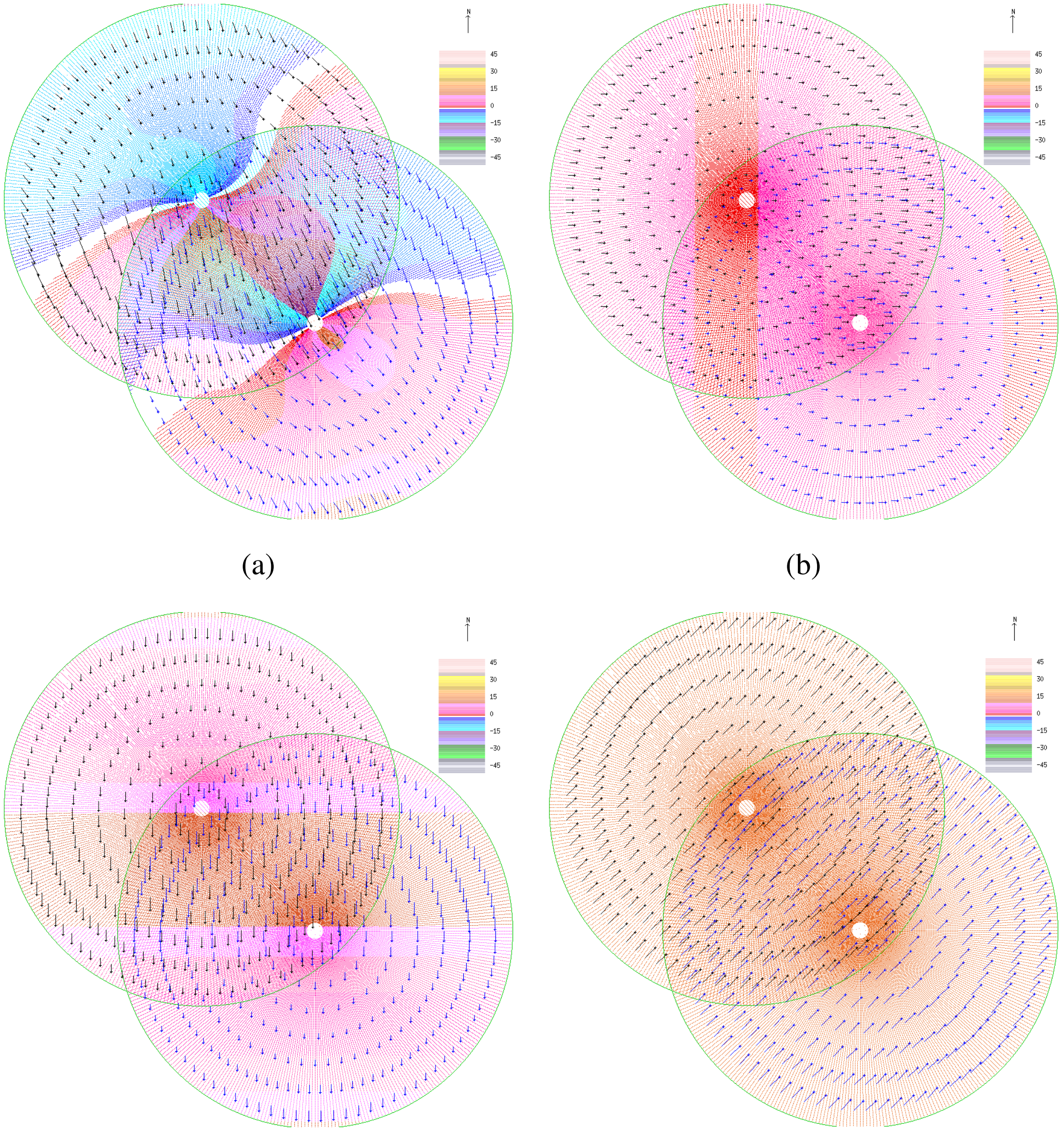
Figure A.37: The single retrieved synthetic velocity of group 2 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

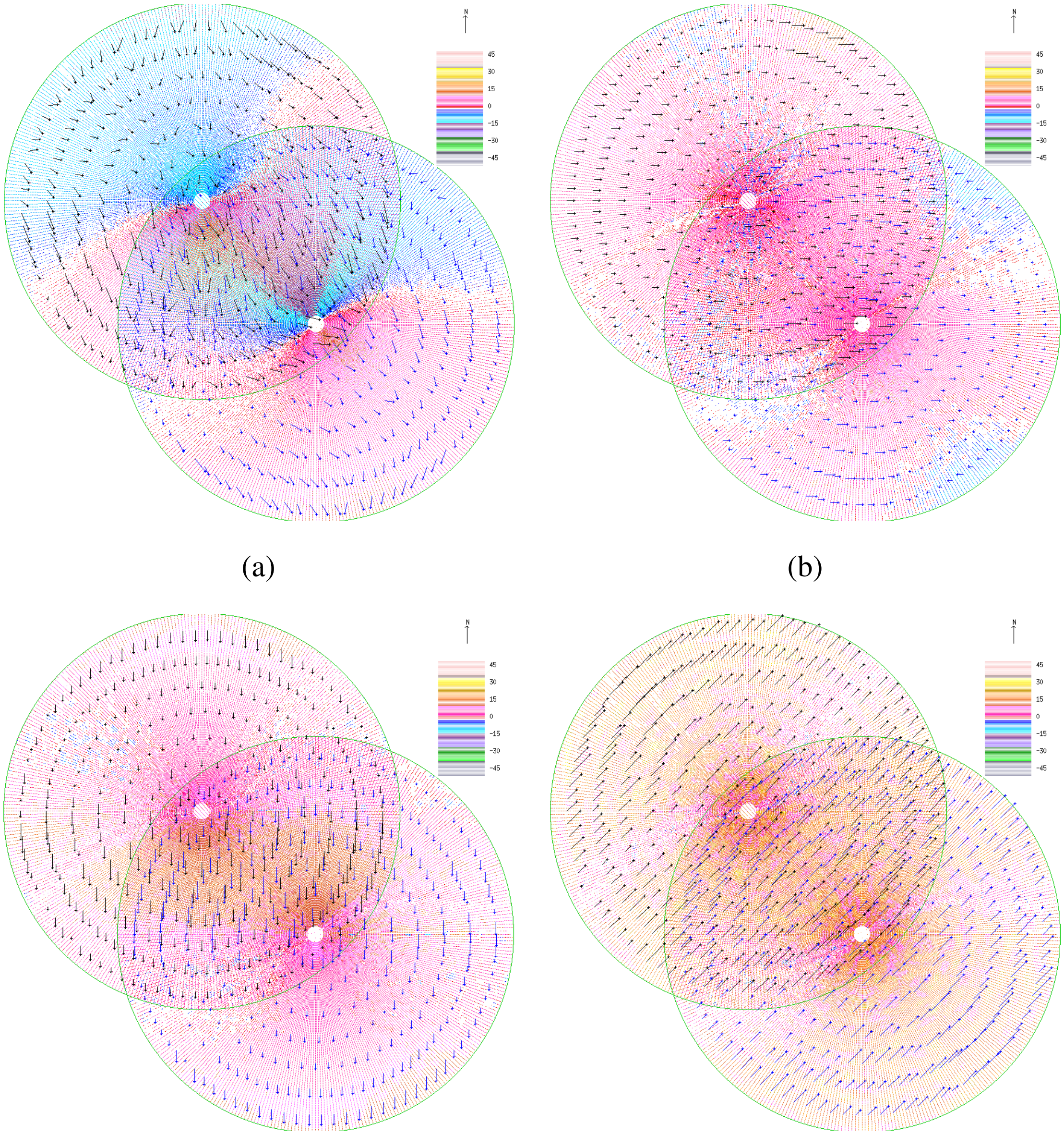
Figure A.38: The dual retrieved synthetic velocity of group 2 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

Figure A.39: The correct synthetic velocity of group 2 at variation level $K = 5$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



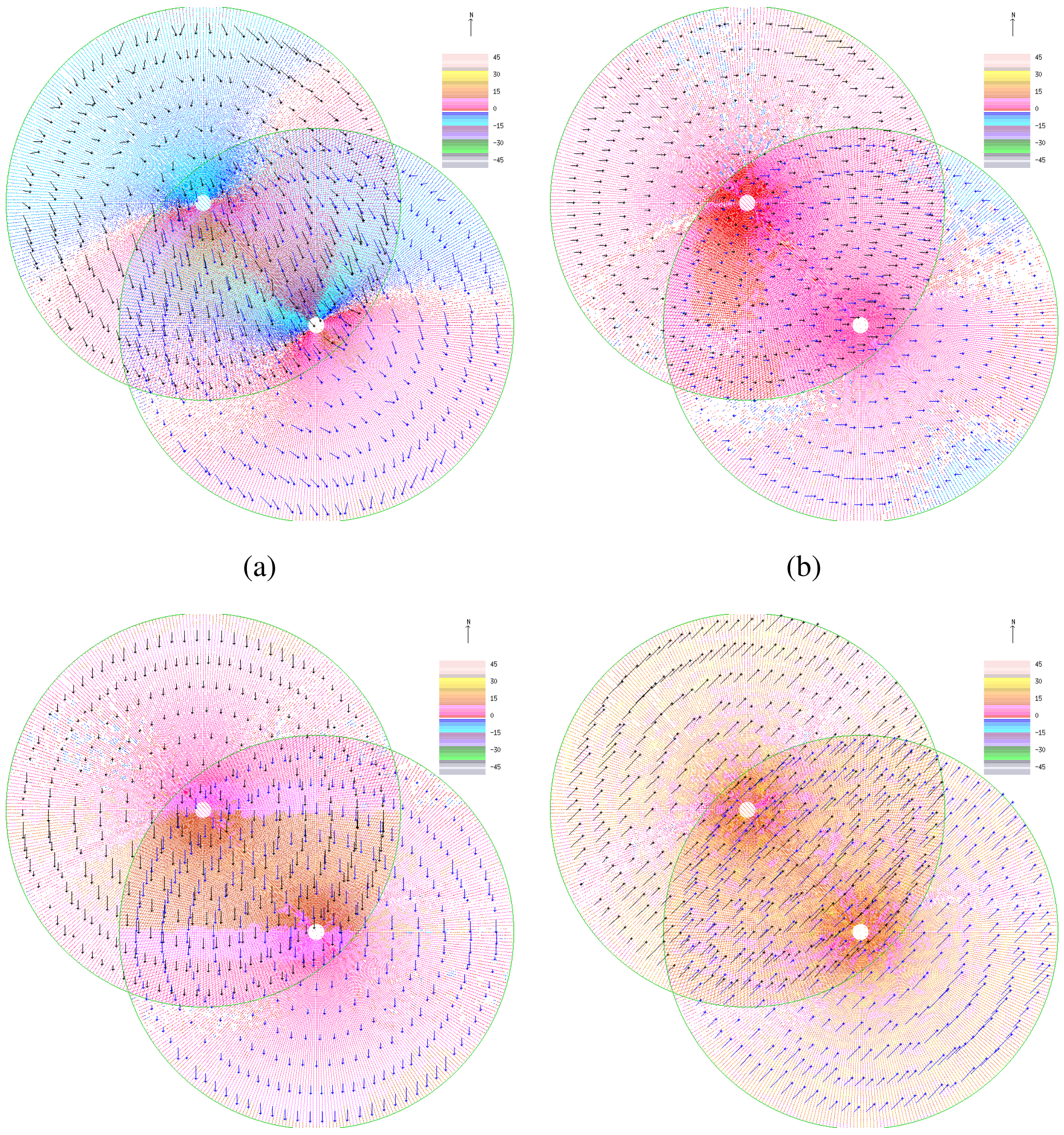
(a)

(b)

(c)

(d)

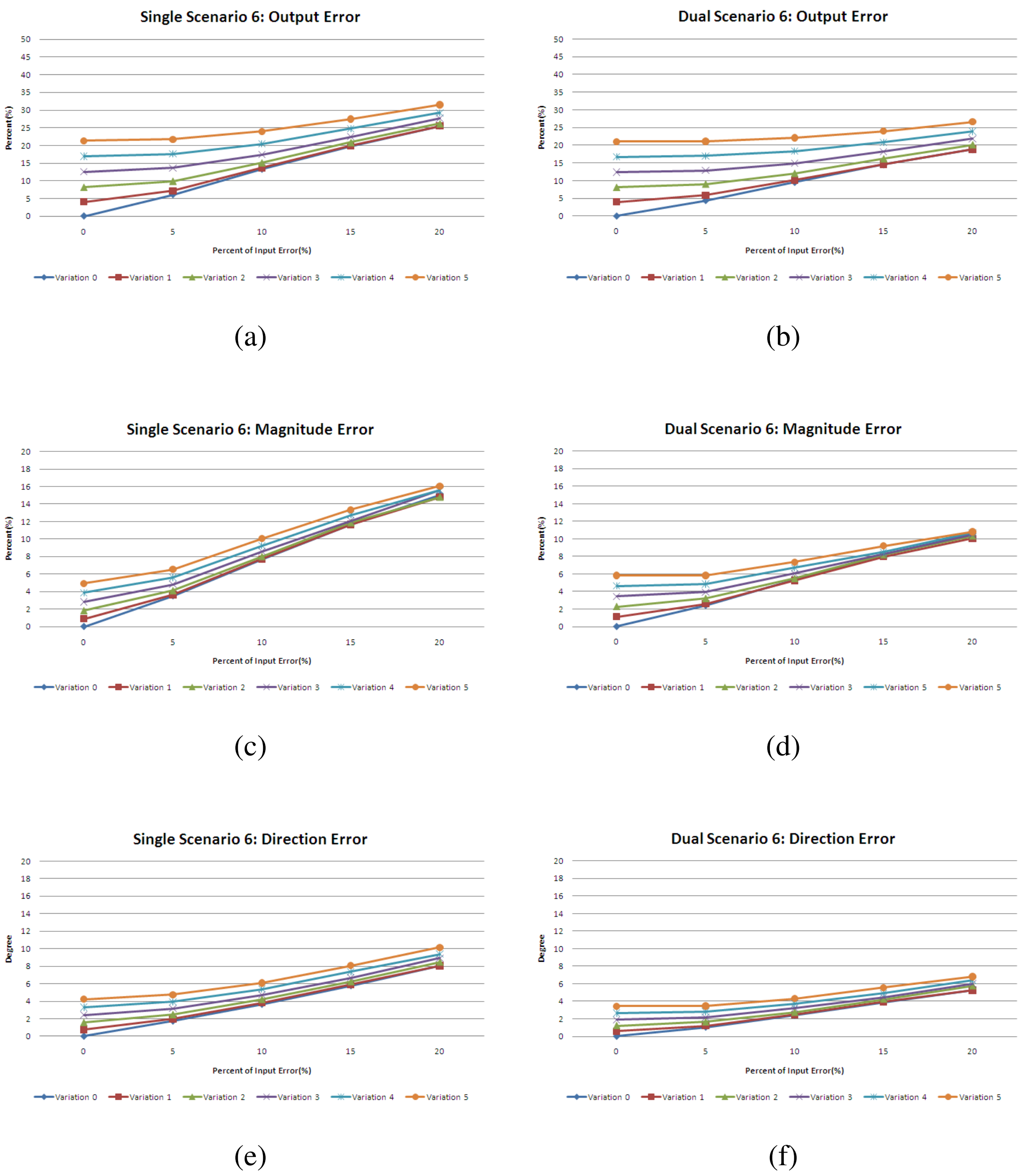
Figure A.40: The single retrieved synthetic velocity of group 2 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

Figure A.41: The dual retrieved synthetic velocity of group 2 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.

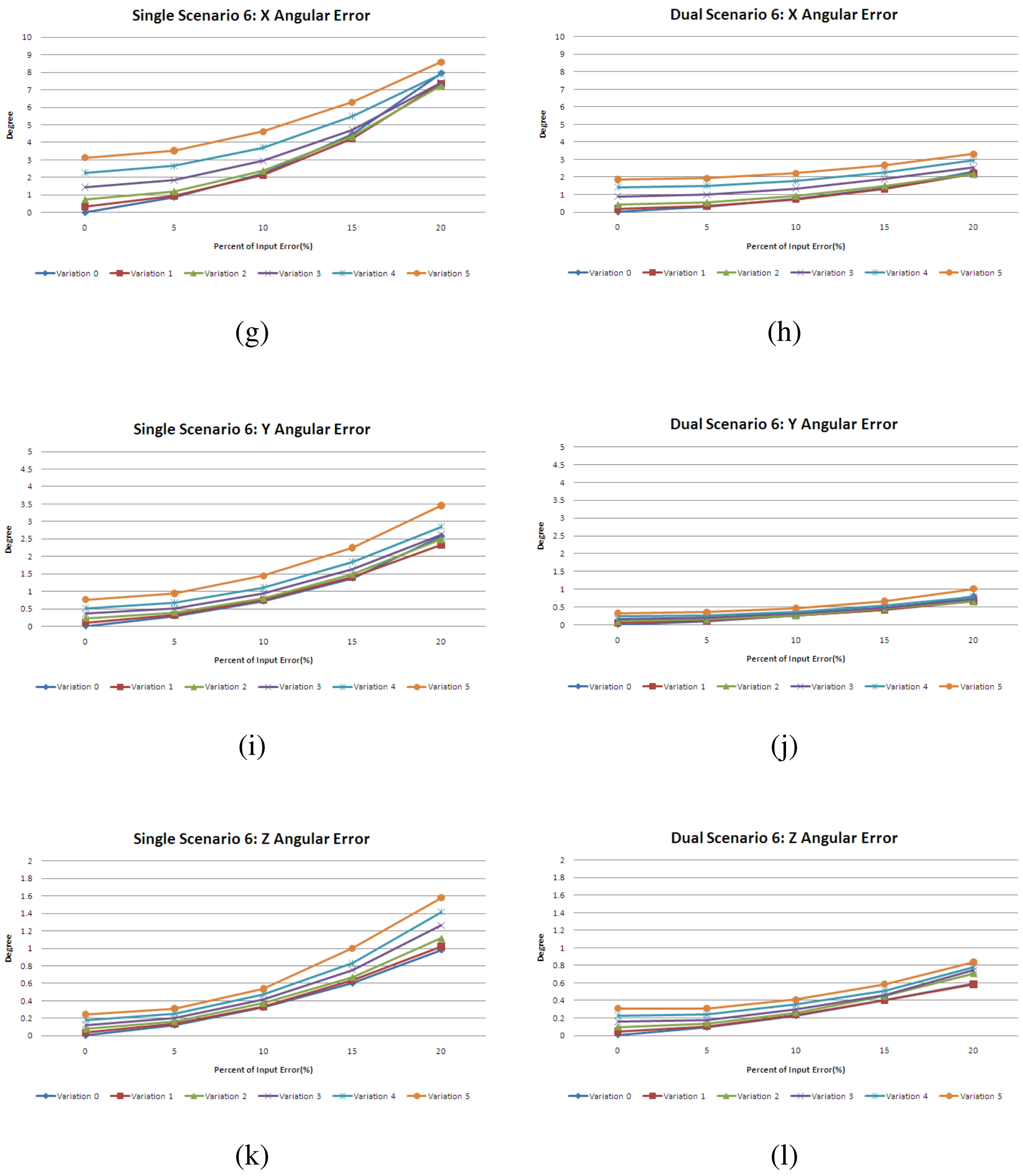


(a) (b)

(c) (d)

(e) (f)

Figure A.42: Result Analysis of Synthetic Data Group 6: (5.0, 10.0, 20.0). The Single Retrieval (a), (c), (e) and Dual Retrieval (b), (d), (f).


Single Scenario 6: X Angular Error

Dual Scenario 6: X Angular Error

(g) (h)

Single Scenario 6: Y Angular Error

Dual Scenario 6: Y Angular Error

(i) (j)

Single Scenario 6: Z Angular Error

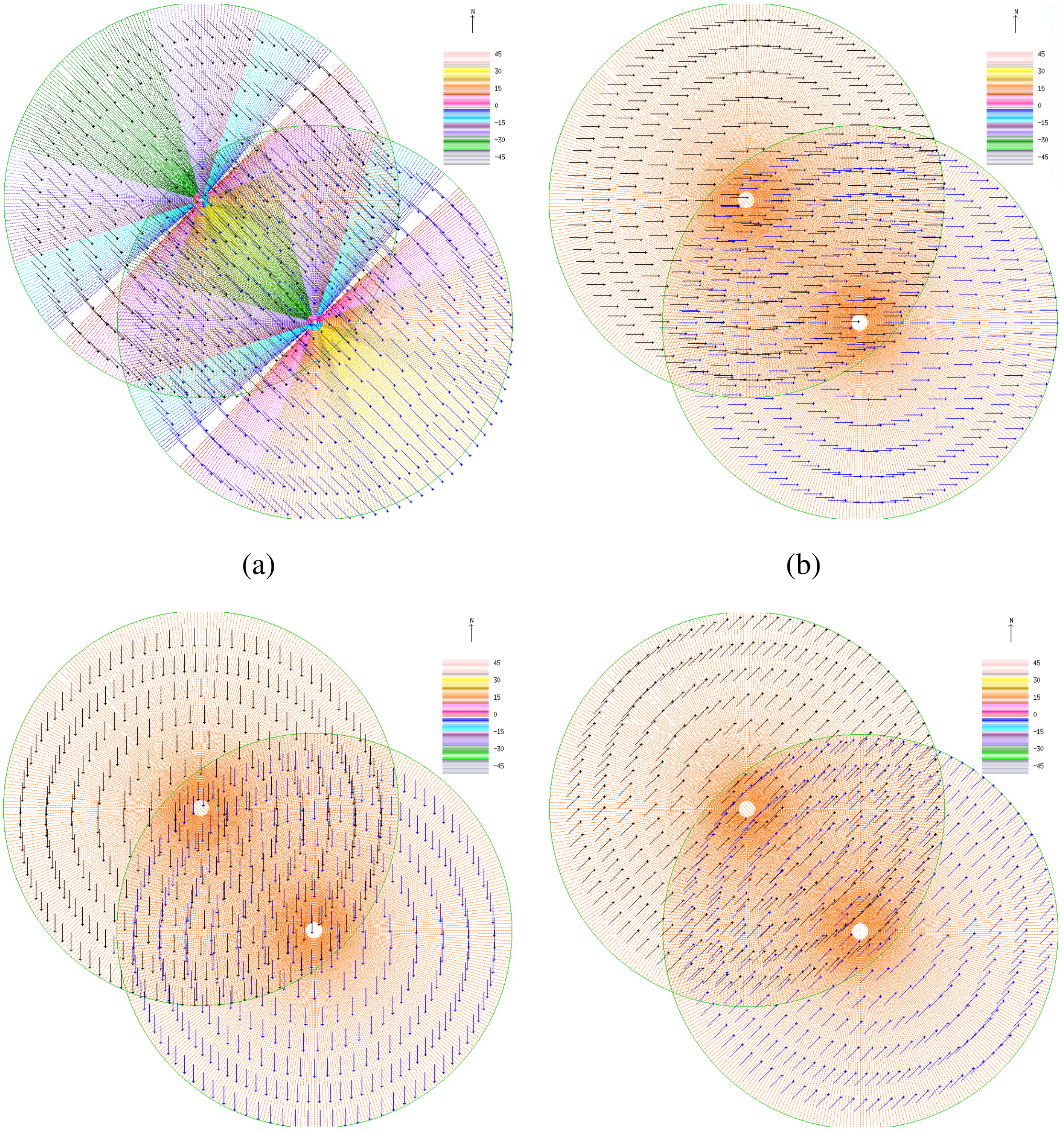
Dual Scenario 6: Z Angular Error

(k) (l)

Figure A.42: Result Analysis of Synthetic Data Group 6: (5.0, 10.0, 20.0). The Single Retrieval (g), (i), (k) and Dual retrieval (h), (j), (l).

Appendix B

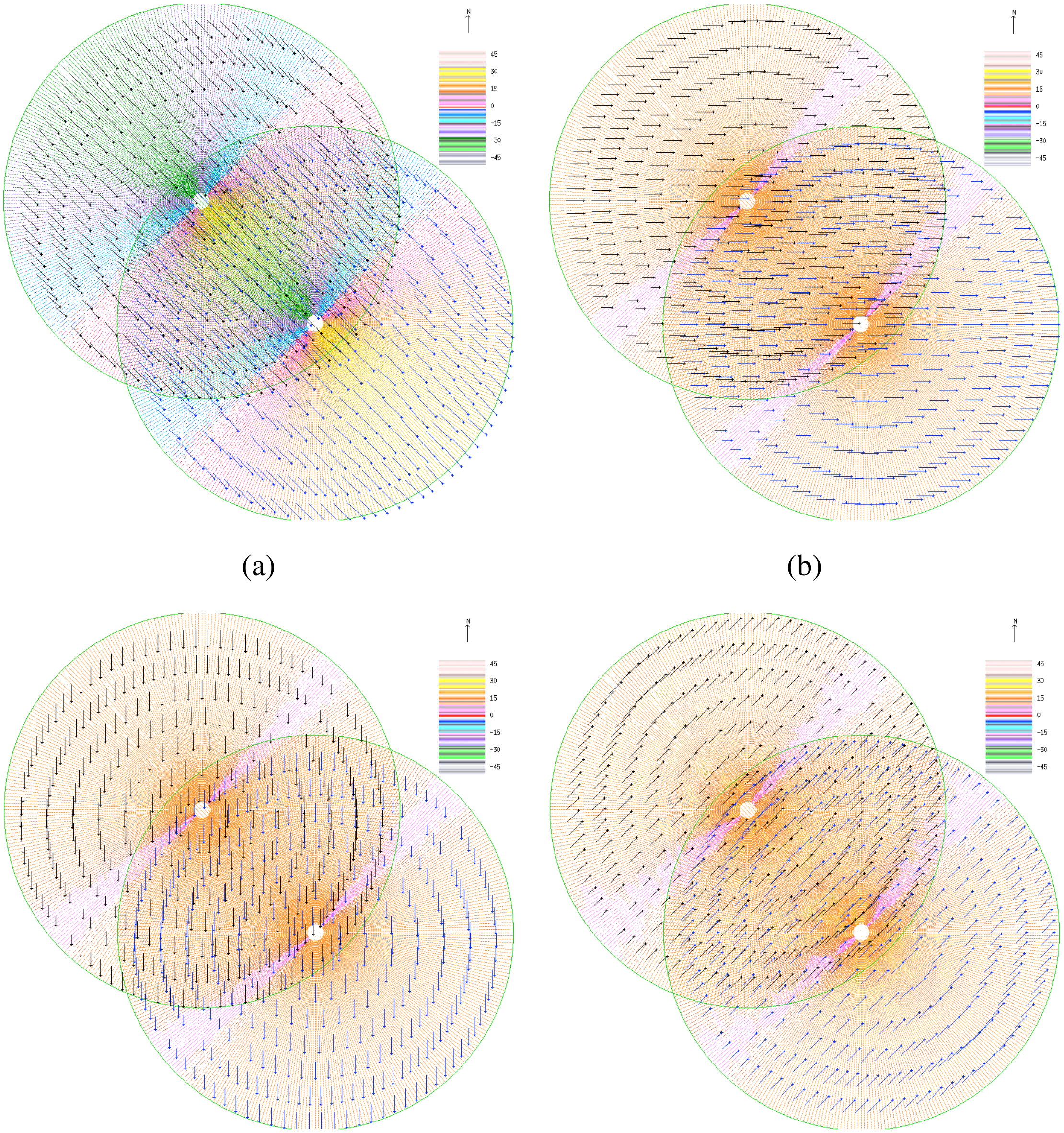
Synthetic Experiment Results for the Regularization Method



(a) (b)

(c) (d)

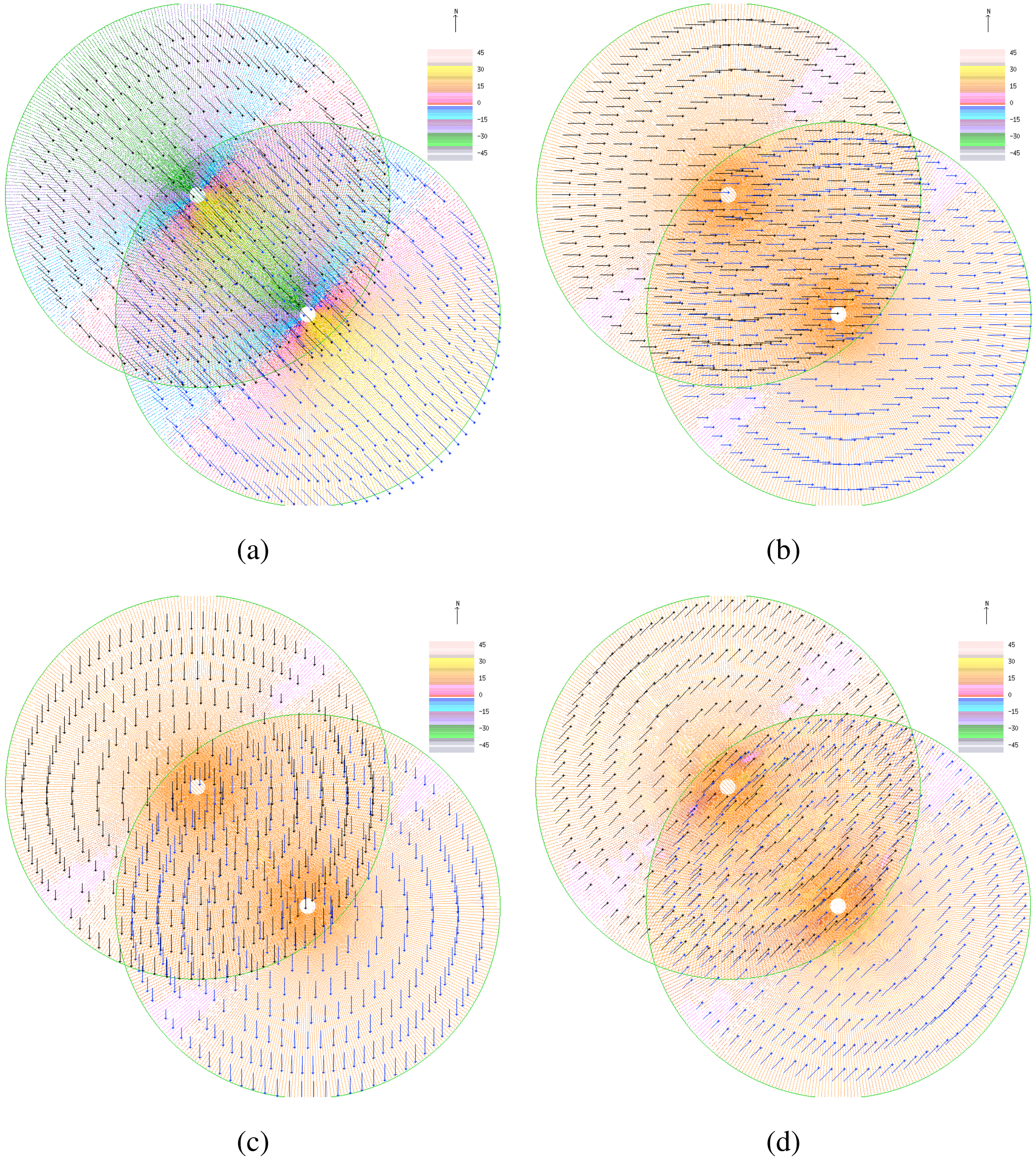
Figure B.1: The correct synthetic velocity of group 1 at variation level $K = 0$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

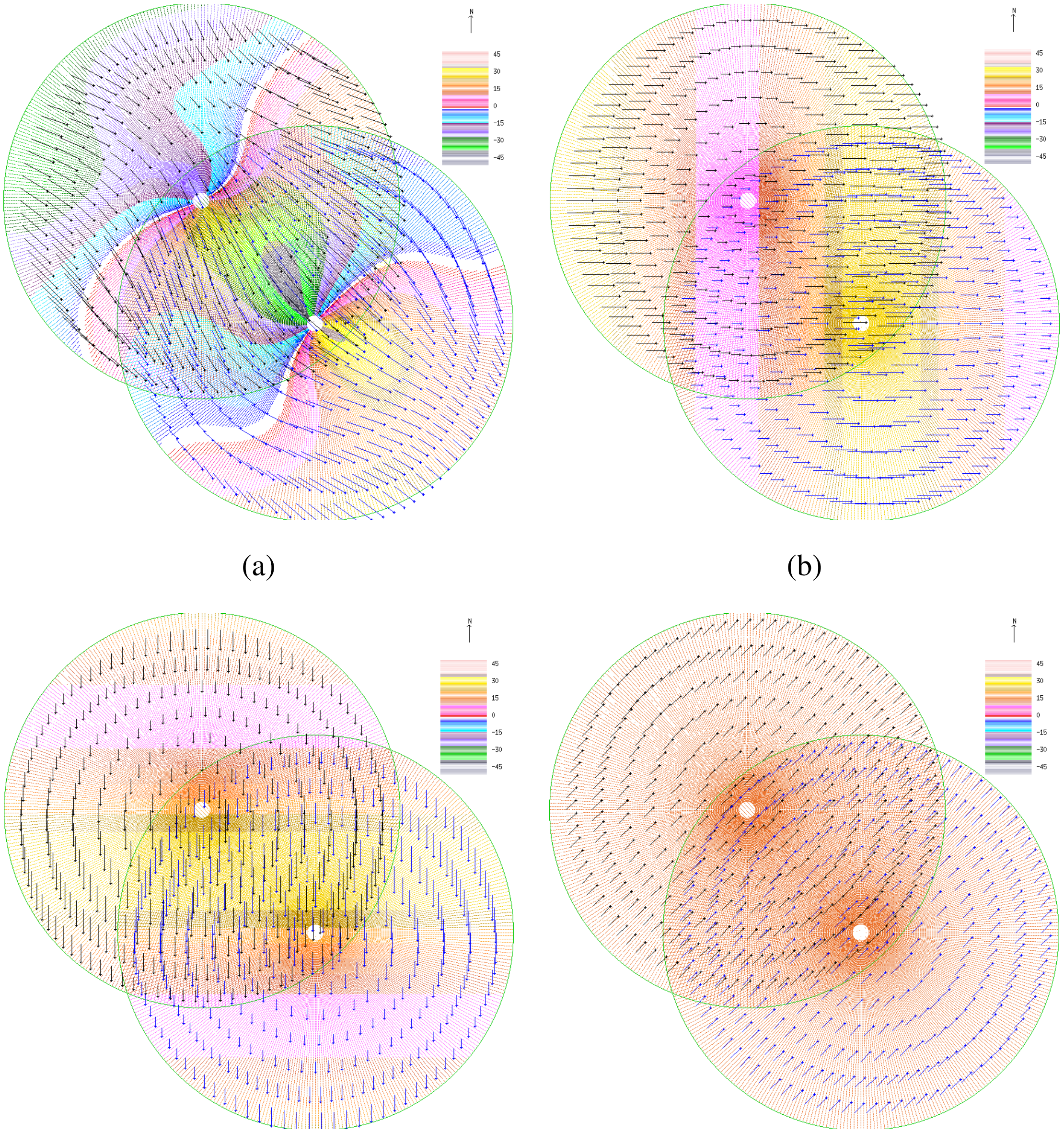
Figure B.2: The single regularization retrieved synthetic velocity of group 1 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

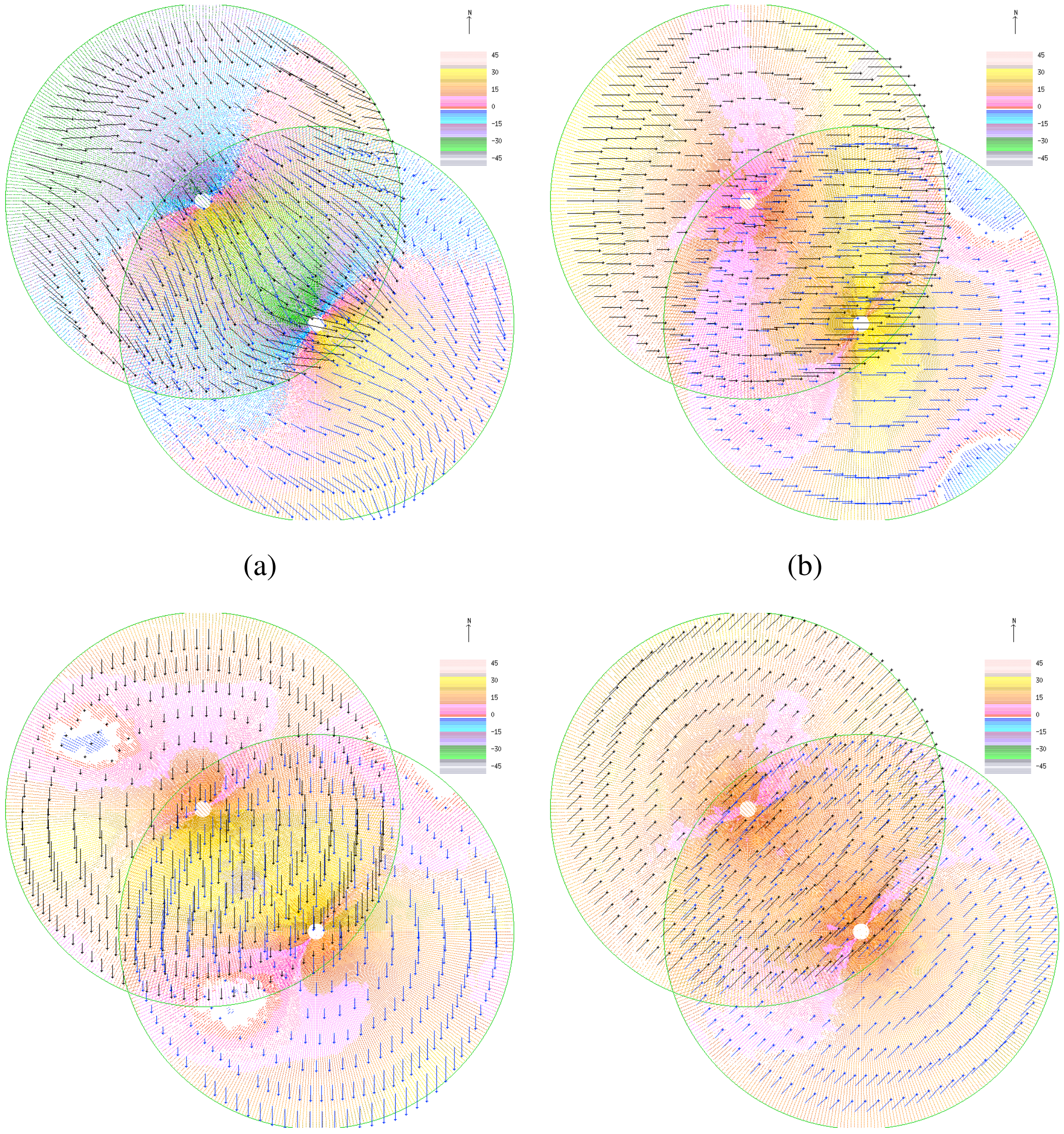
Figure B.3: The dual regularization retrieved synthetic velocity of group 1 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

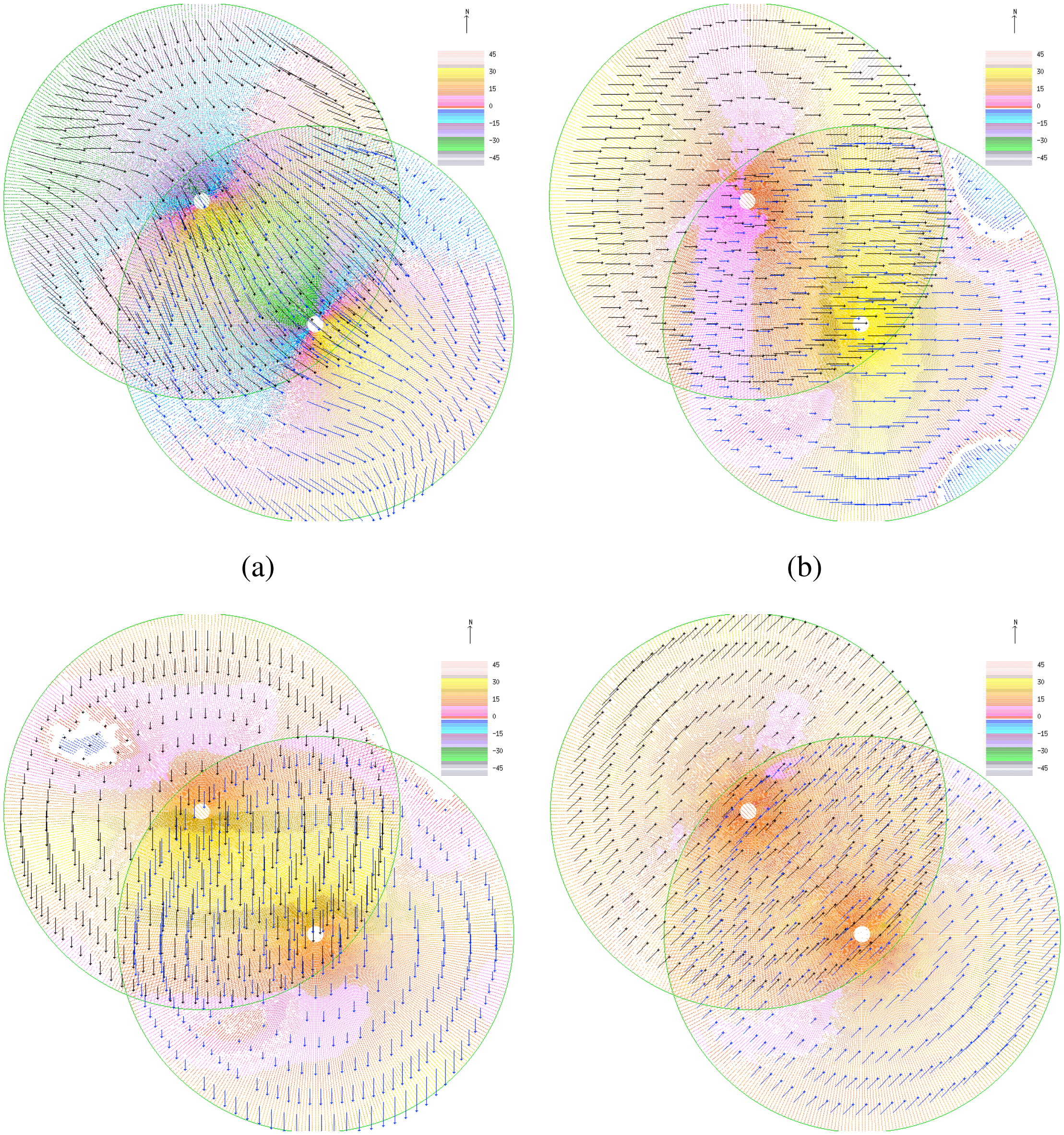
Figure B.4: The correct synthetic velocity of group 1 at variation level $K = 5$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

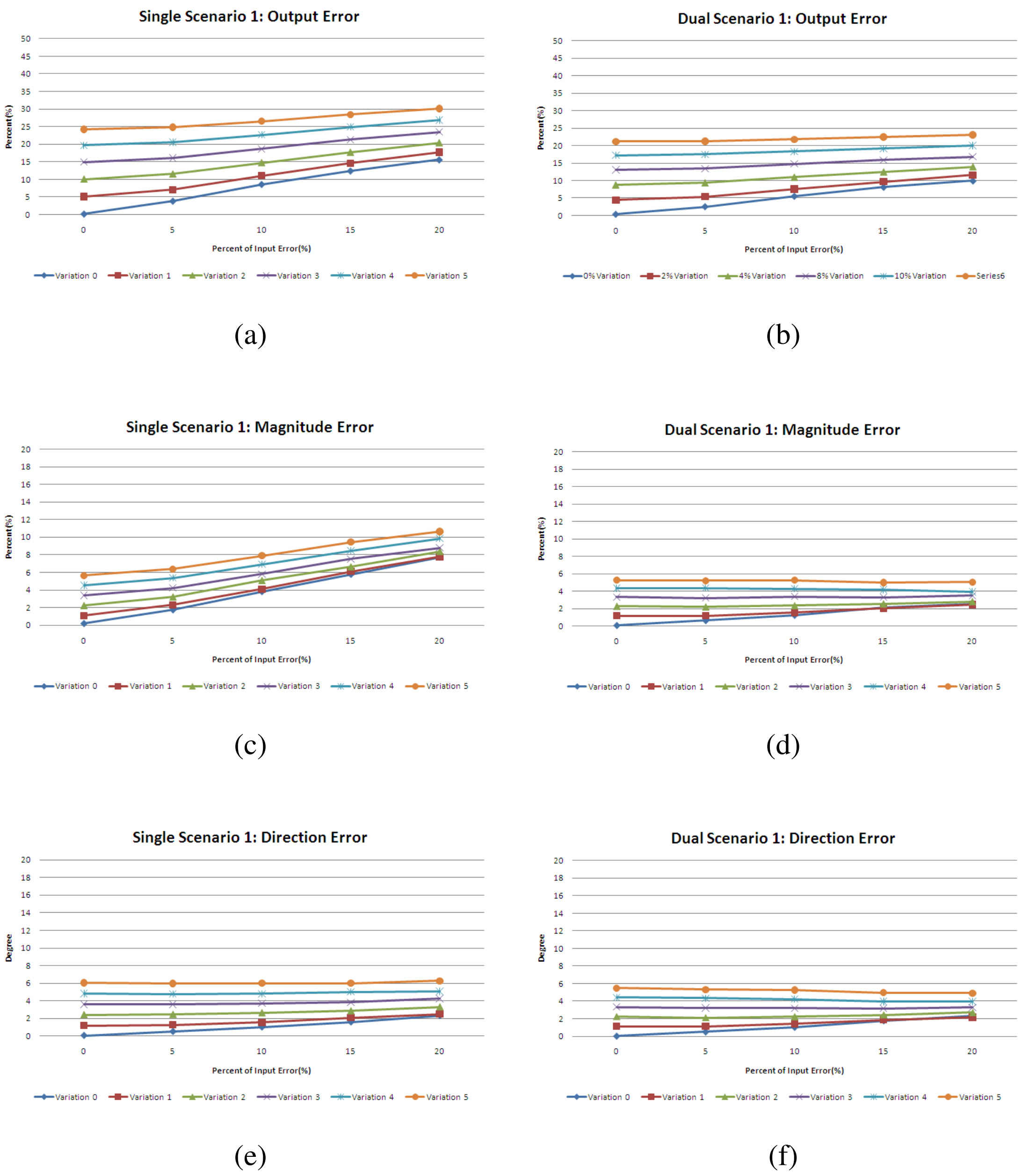
Figure B.5: The single regularization retrieved synthetic velocity of group 1 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

Figure B.6: The dual regularization retrieved synthetic velocity of group 1 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.

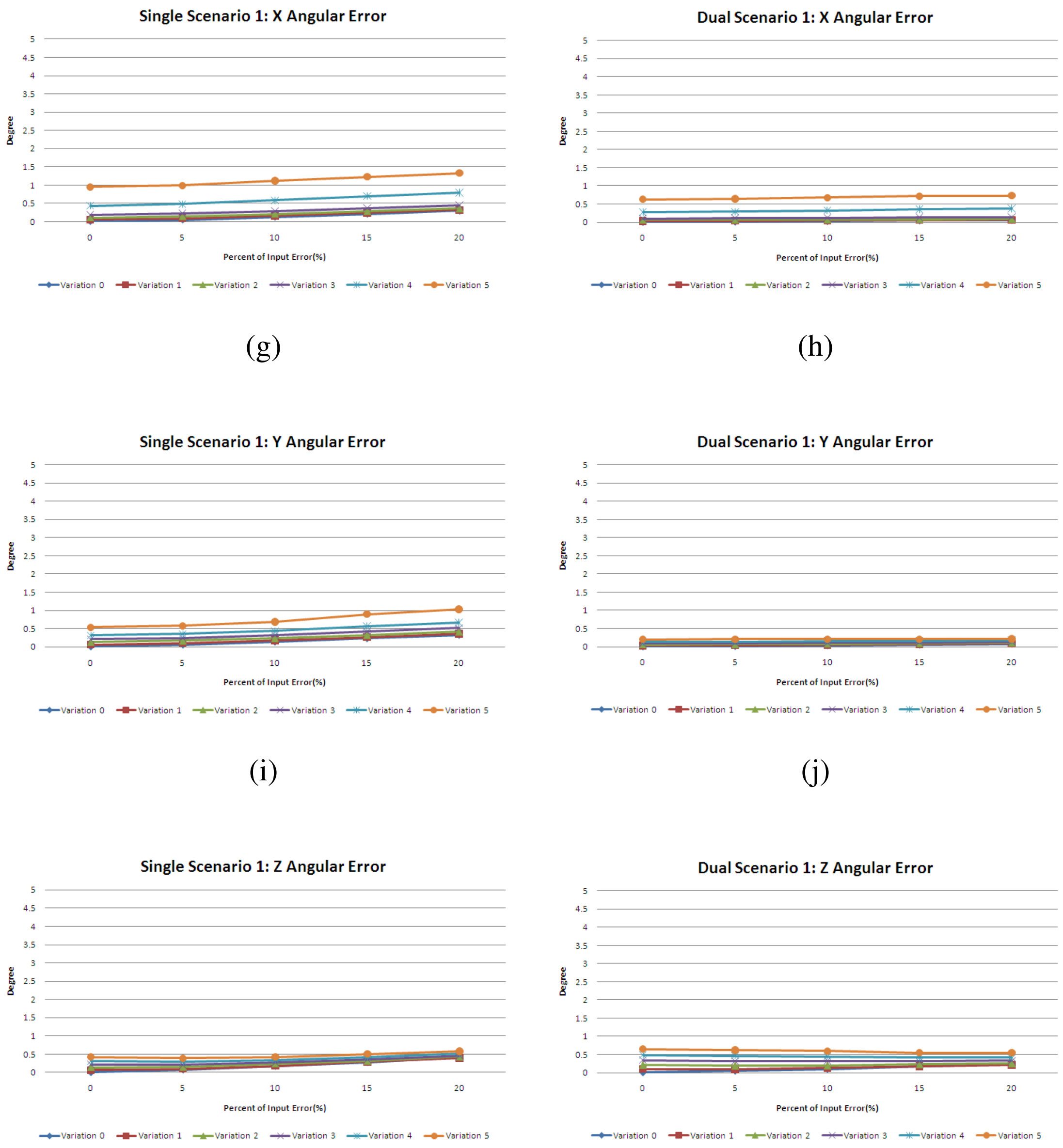


(a) (b)

(c) (d)

(e) (f)

Figure B.7: Result Analysis of Synthetic Data Group 1 using regularization method: (20.0, 20.0, 20.0). The Single Retrieval (a), (c), (e) and Dual Retrieval (b), (d), (f).


Single Scenario 1: X Angular Error

Dual Scenario 1: X Angular Error

Single Scenario 1: Y Angular Error

Dual Scenario 1: Y Angular Error

Single Scenario 1: Z Angular Error

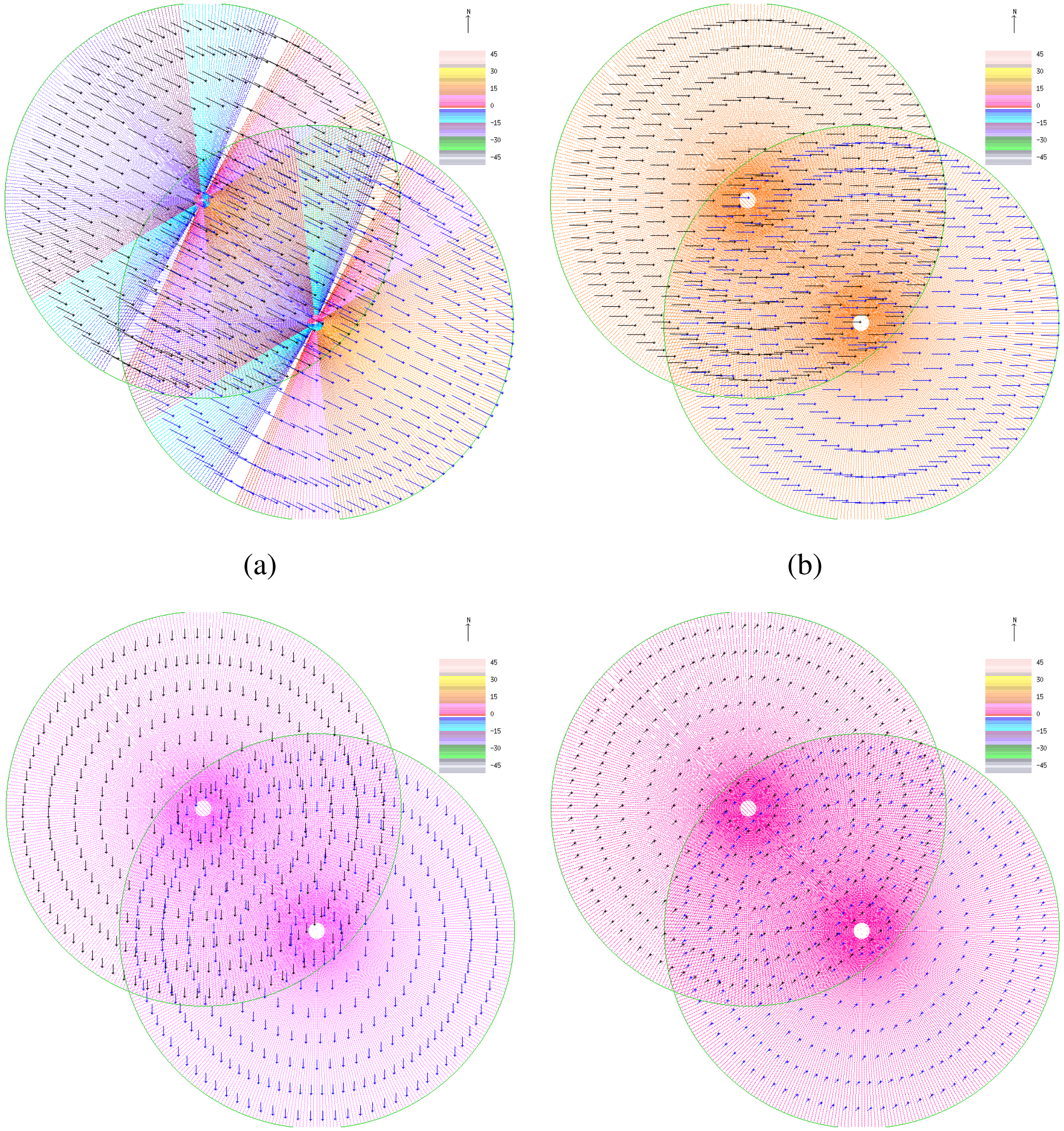
Dual Scenario 1: Z Angular Error

(g) (h)

(i) (j)

(k) (l)

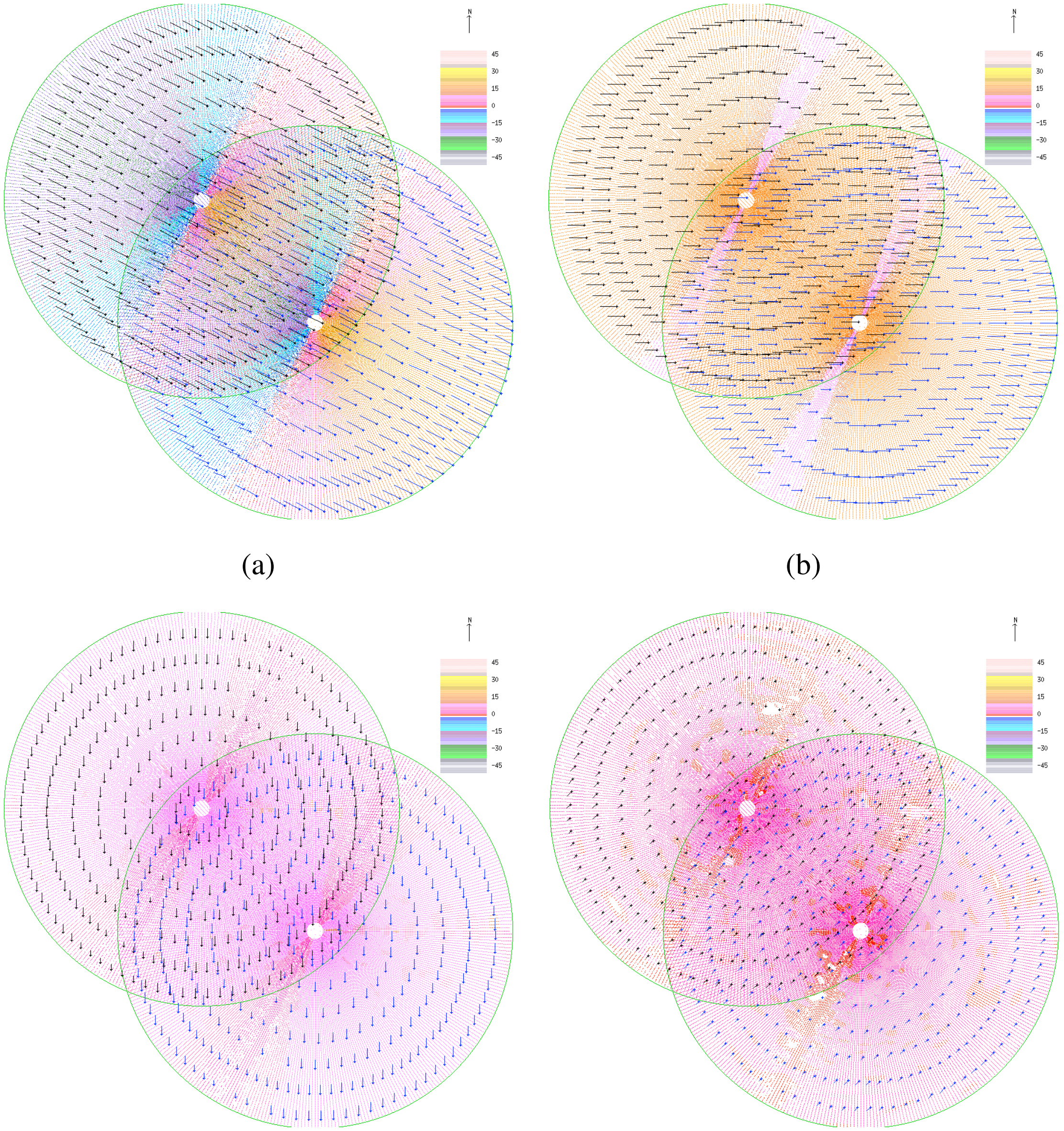
Figure B.7: Result Analysis of Synthetic Data Group 1 using regularization method: (20.0, 20.0, 20.0). The Single Retrieval (g), (i), (k) and Dual retrieval (h), (j), (l).



(a) (b)

(c) (d)

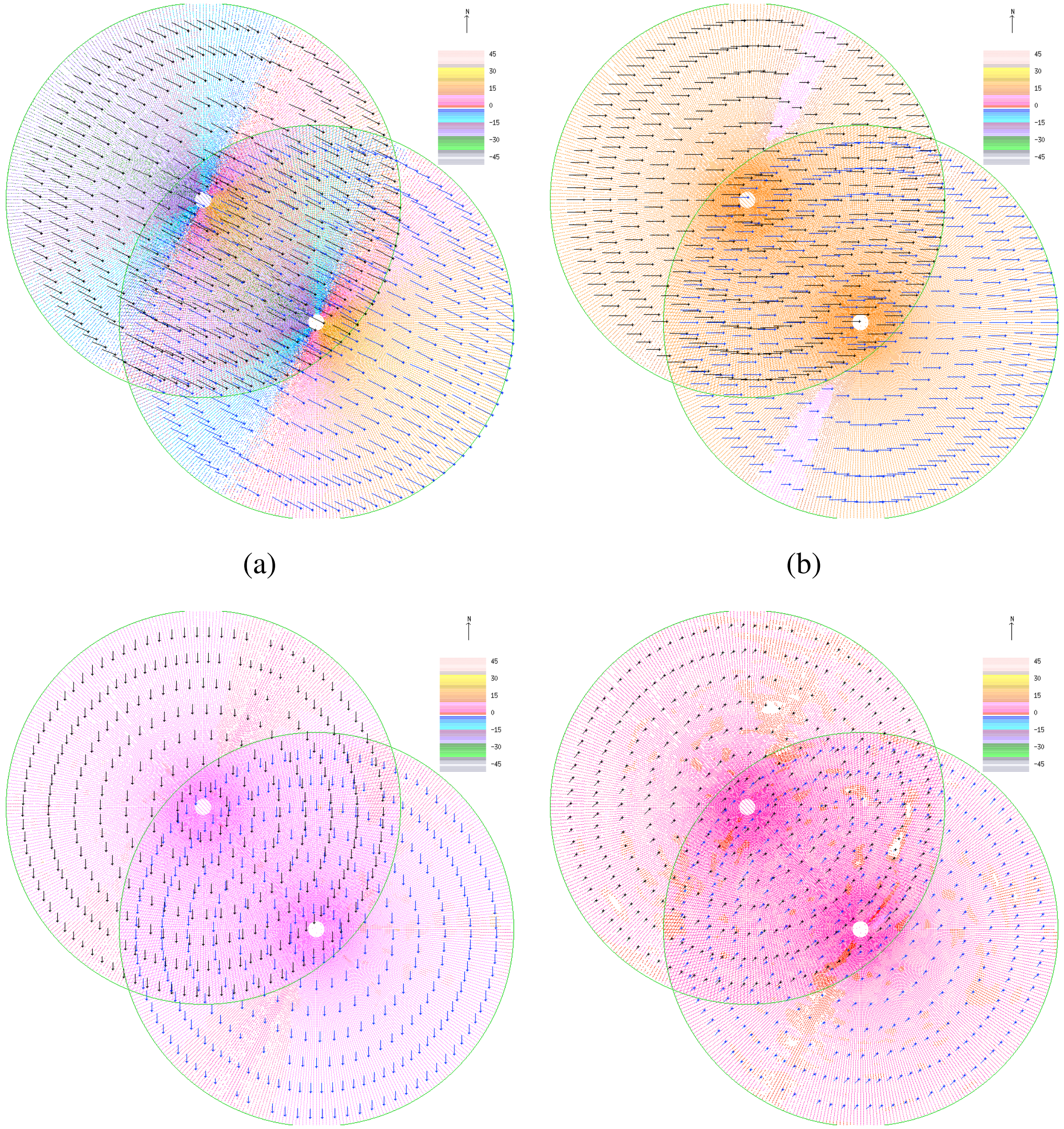
Figure B.8: The correct synthetic velocity of group 2 at variation level $K = 0$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

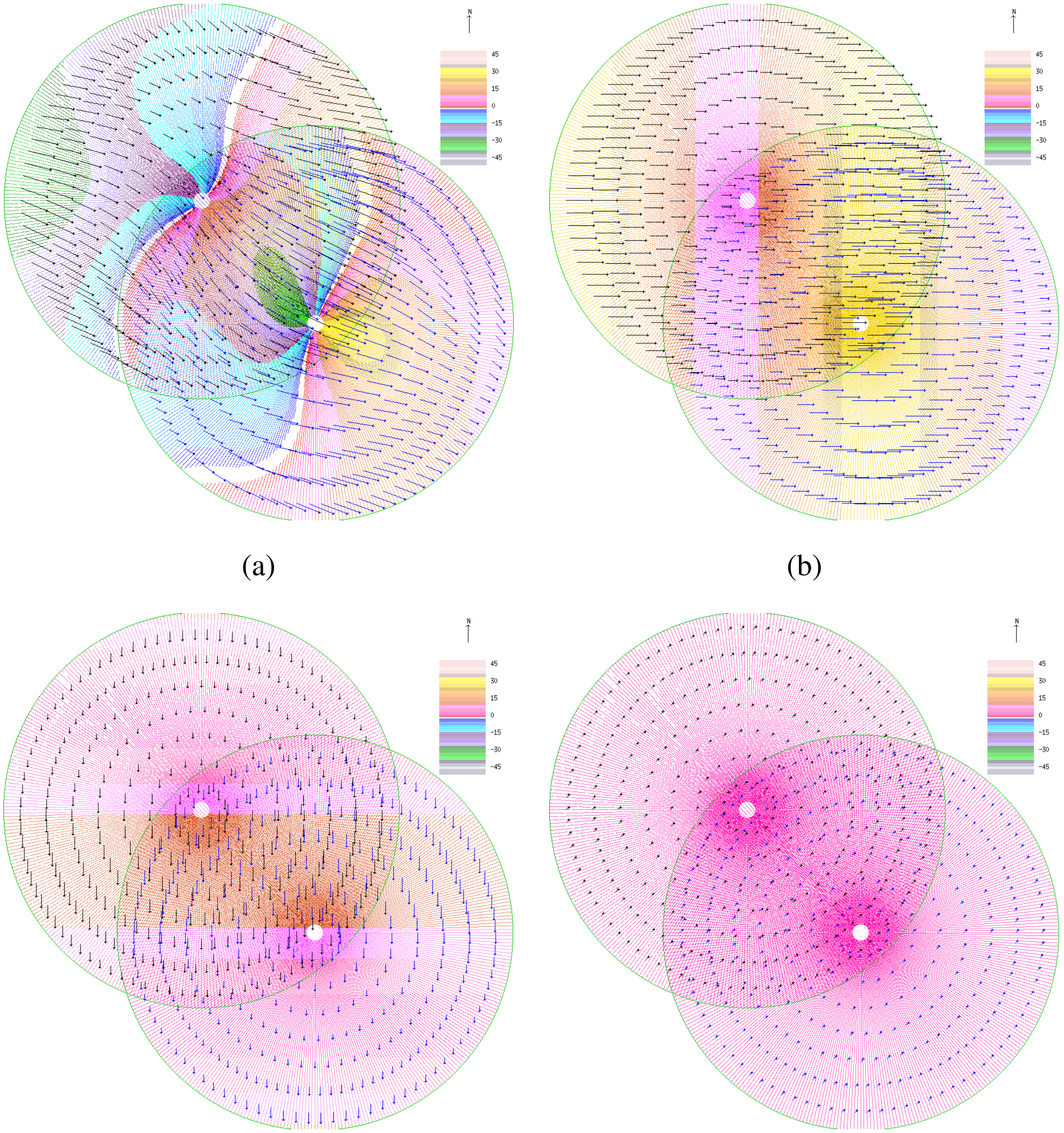
Figure B.9: The single regularization retrieved synthetic velocity of group 2 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

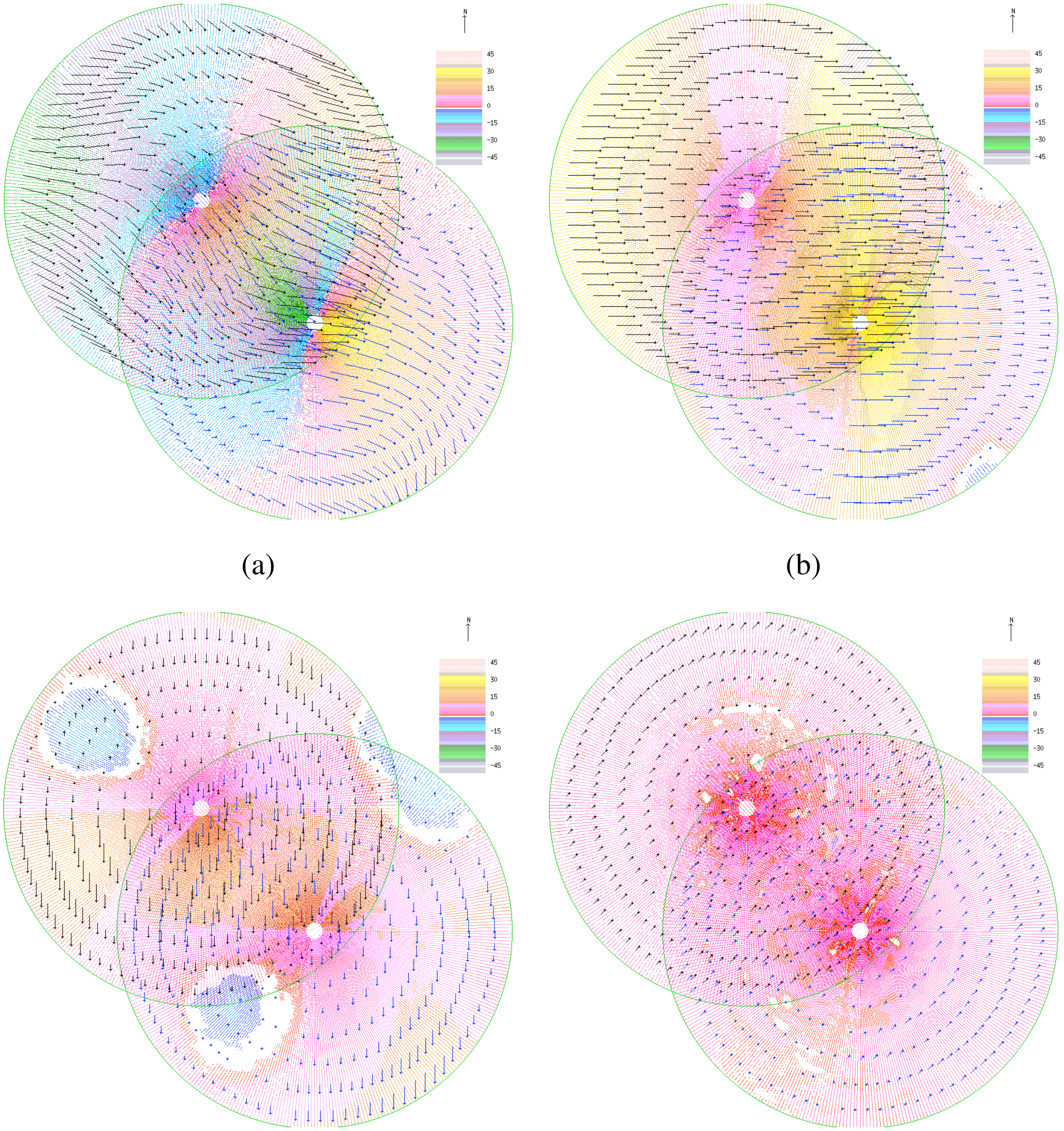
Figure B.10: The dual regularization retrieved synthetic velocity of group 2 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

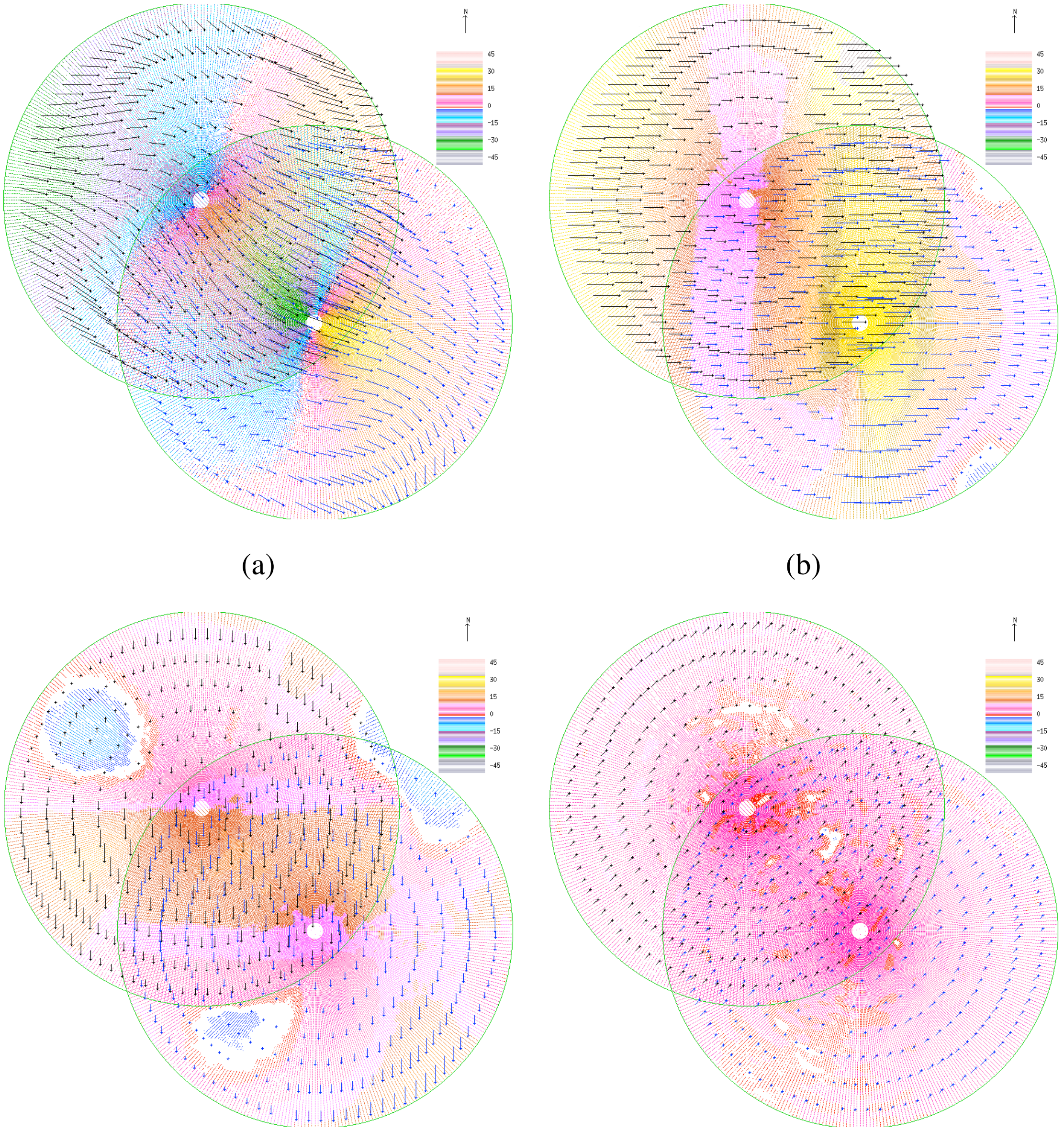
Figure B.11: The correct synthetic velocity of group 2 at variation level $K = 5$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

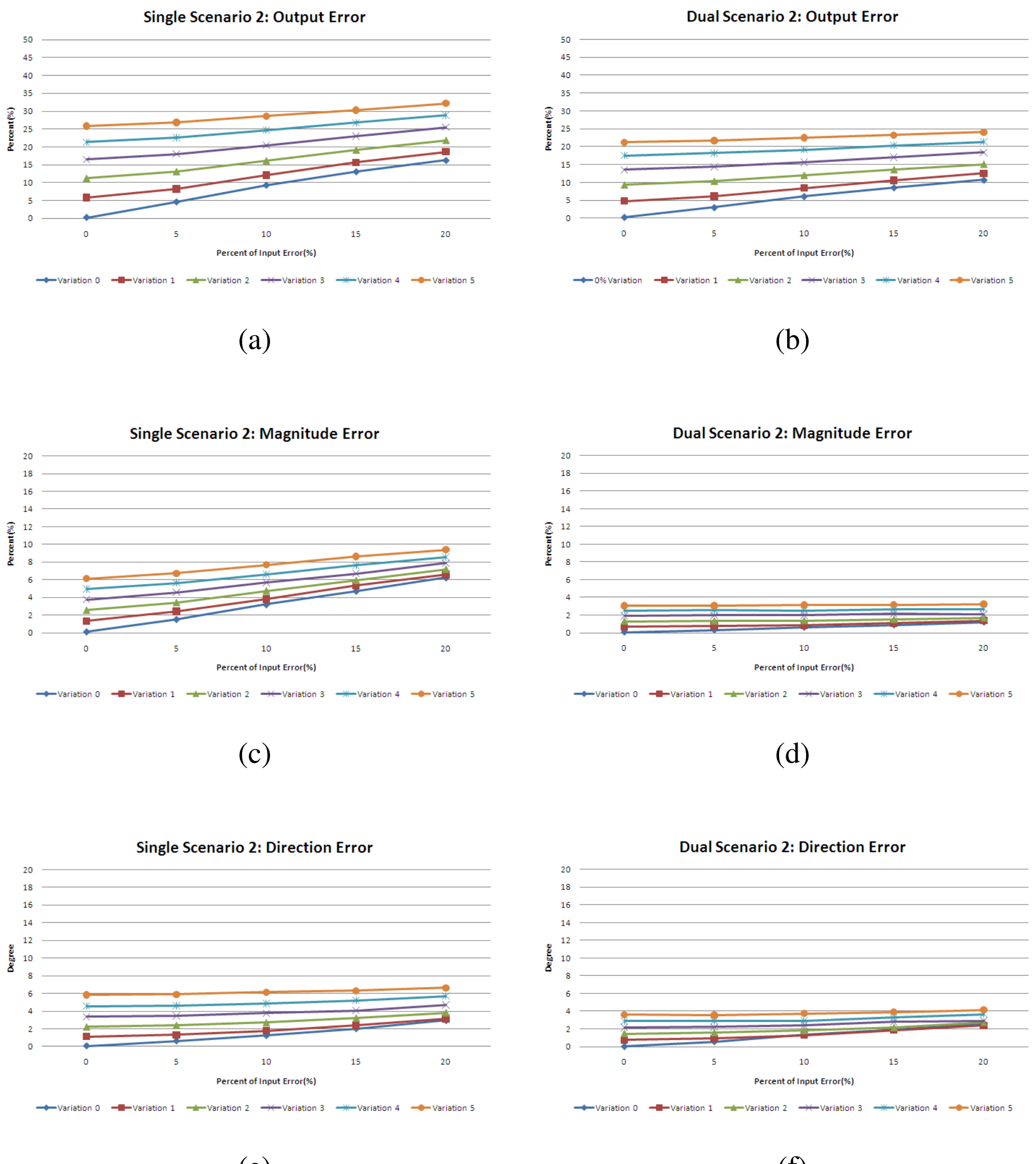
Figure B.12: The single regularization retrieved synthetic velocity of group 2 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

Figure B.13: The dual regularization retrieved synthetic velocity of group 2 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



Single Scenario 2: Output Error

Dual Scenario 2: Output Error

Single Scenario 2: Magnitude Error

Dual Scenario 2: Magnitude Error

Single Scenario 2: Direction Error

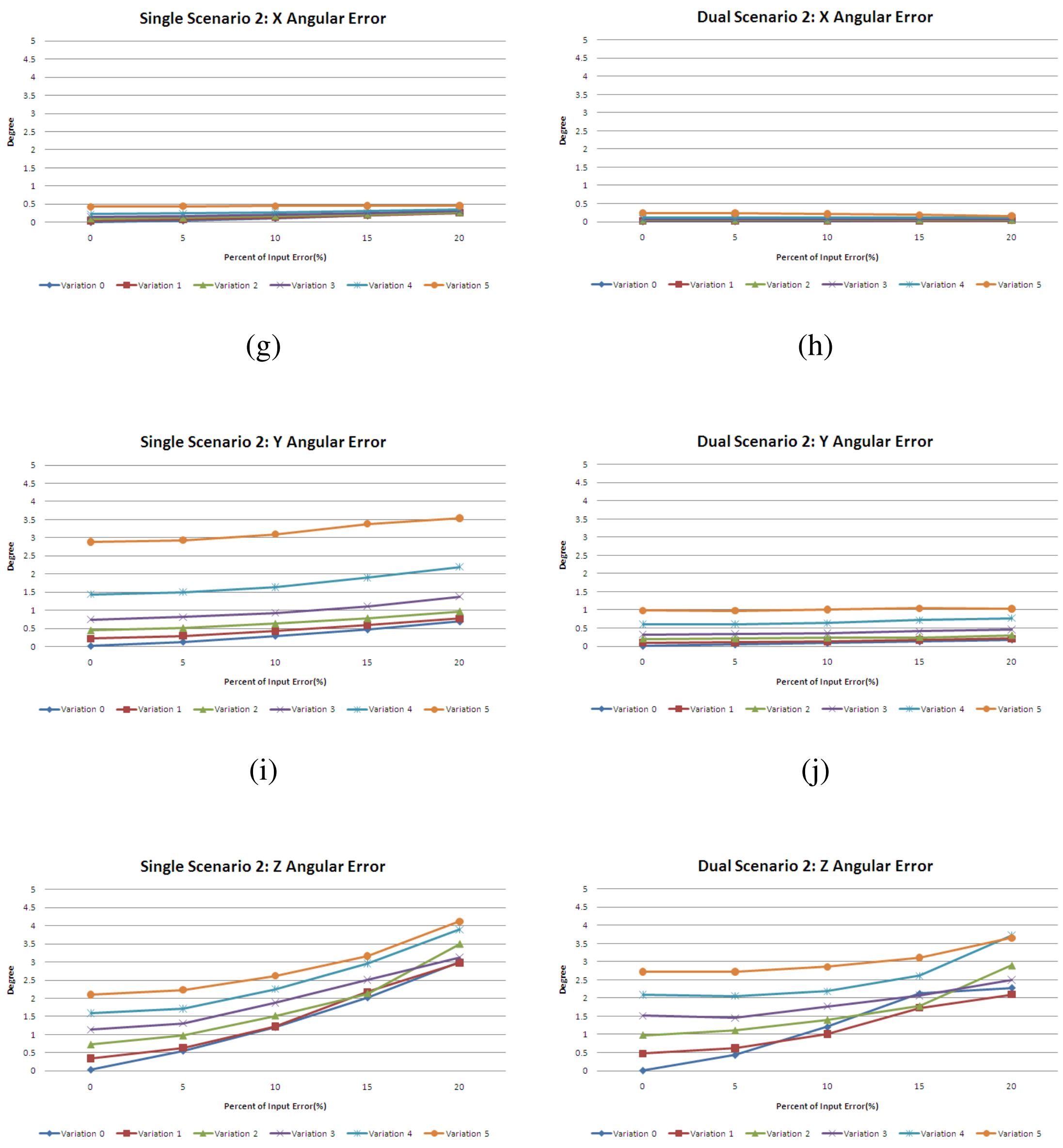
Dual Scenario 2: Direction Error

(a) (b)

(c) (d)

(e) (f)

Figure B.14: Result Analysis of Synthetic Data Group 2 using regularization method: (20.0, 10.0, 5.0). The Single Retrieval (a), (c), (e) and Dual Retrieval (b), (d), (f).

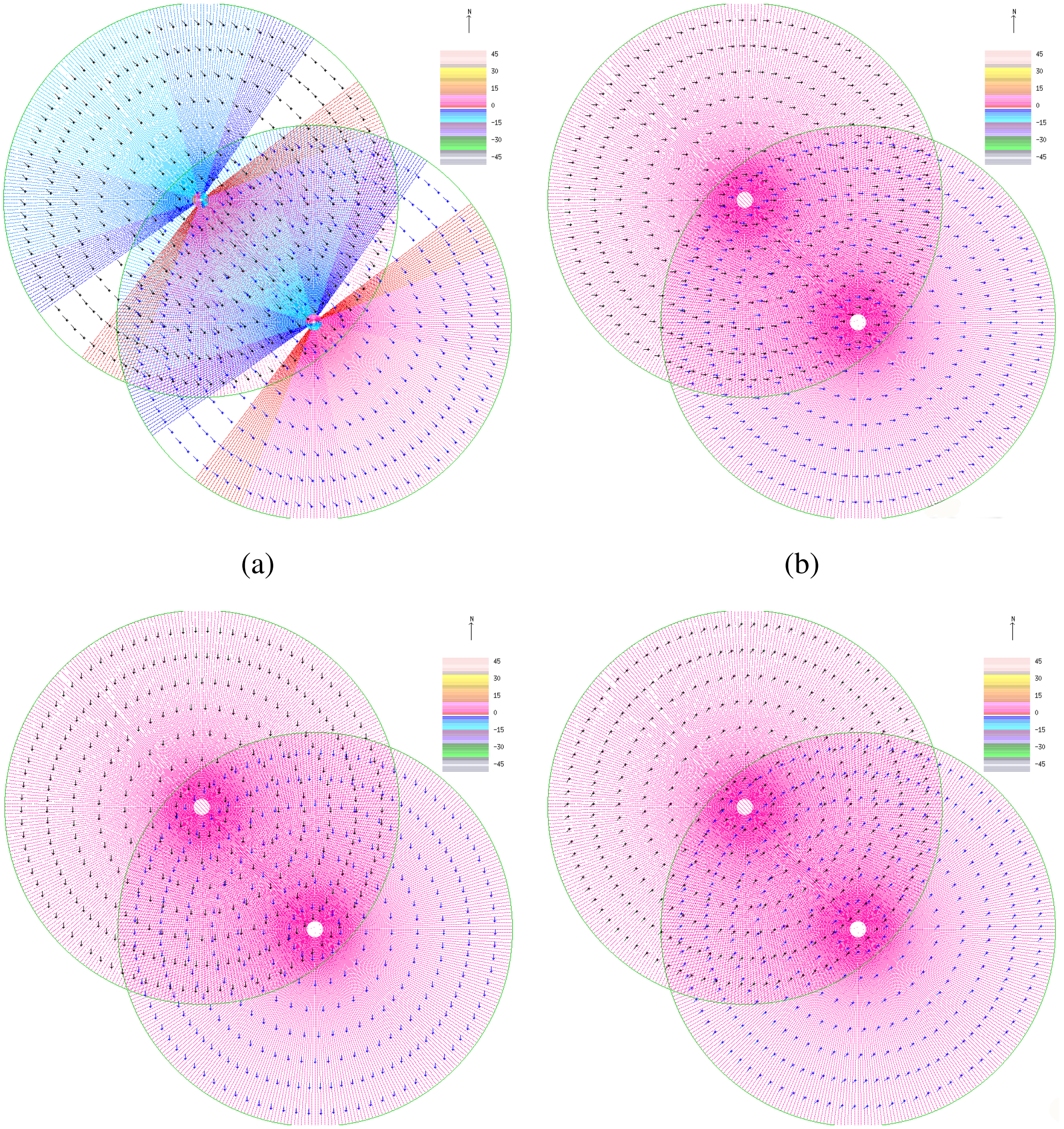


(g) (h)

(i) (j)

(k) (l)

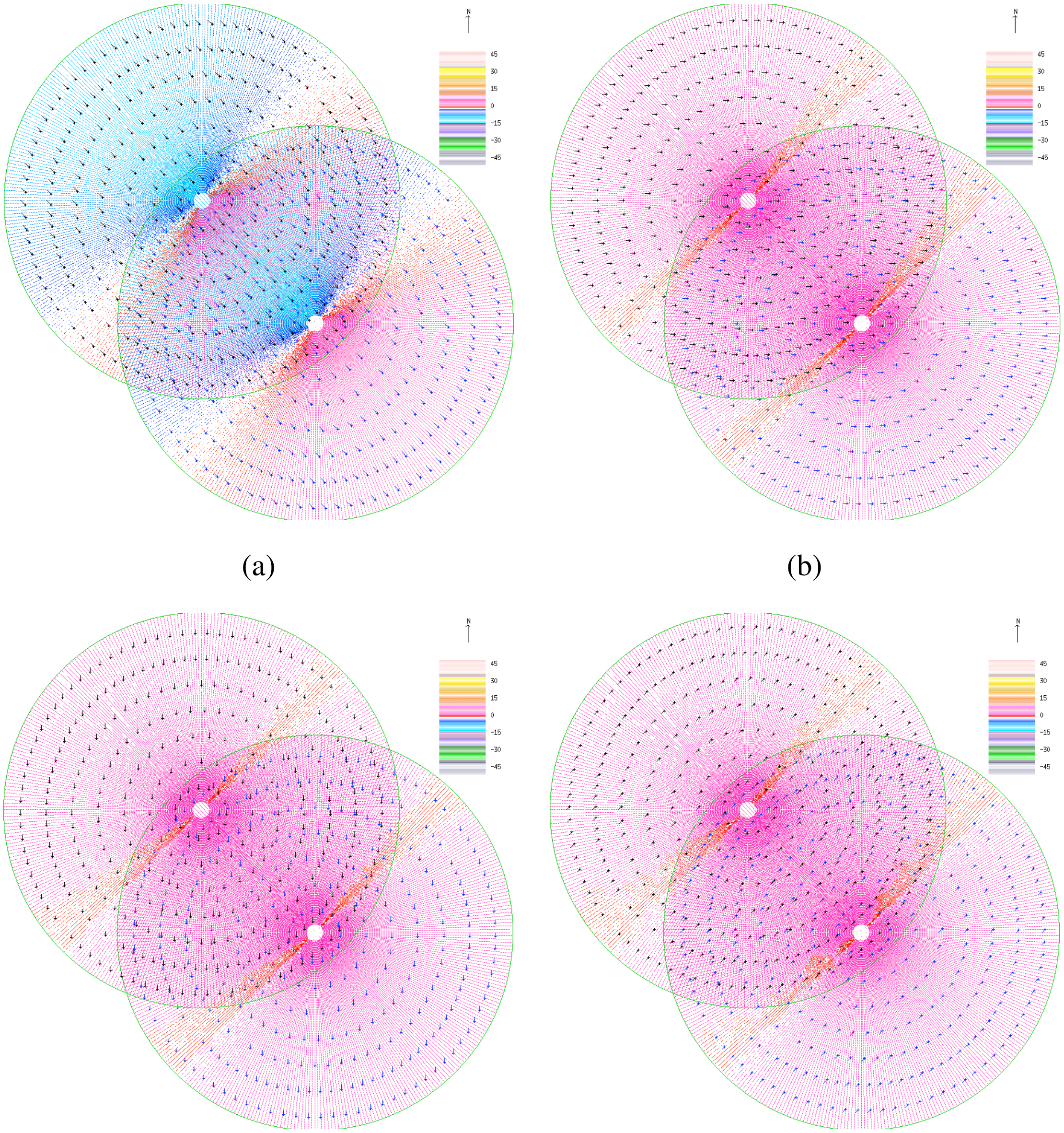
Figure B.14: Result Analysis of Synthetic Data Group 2 using regularization method: (20.0, 10.0, 5.0). The Single Retrieval (g), (i), (k) and Dual retrieval (h), (j), (l).



(a) (b)

(c) (d)

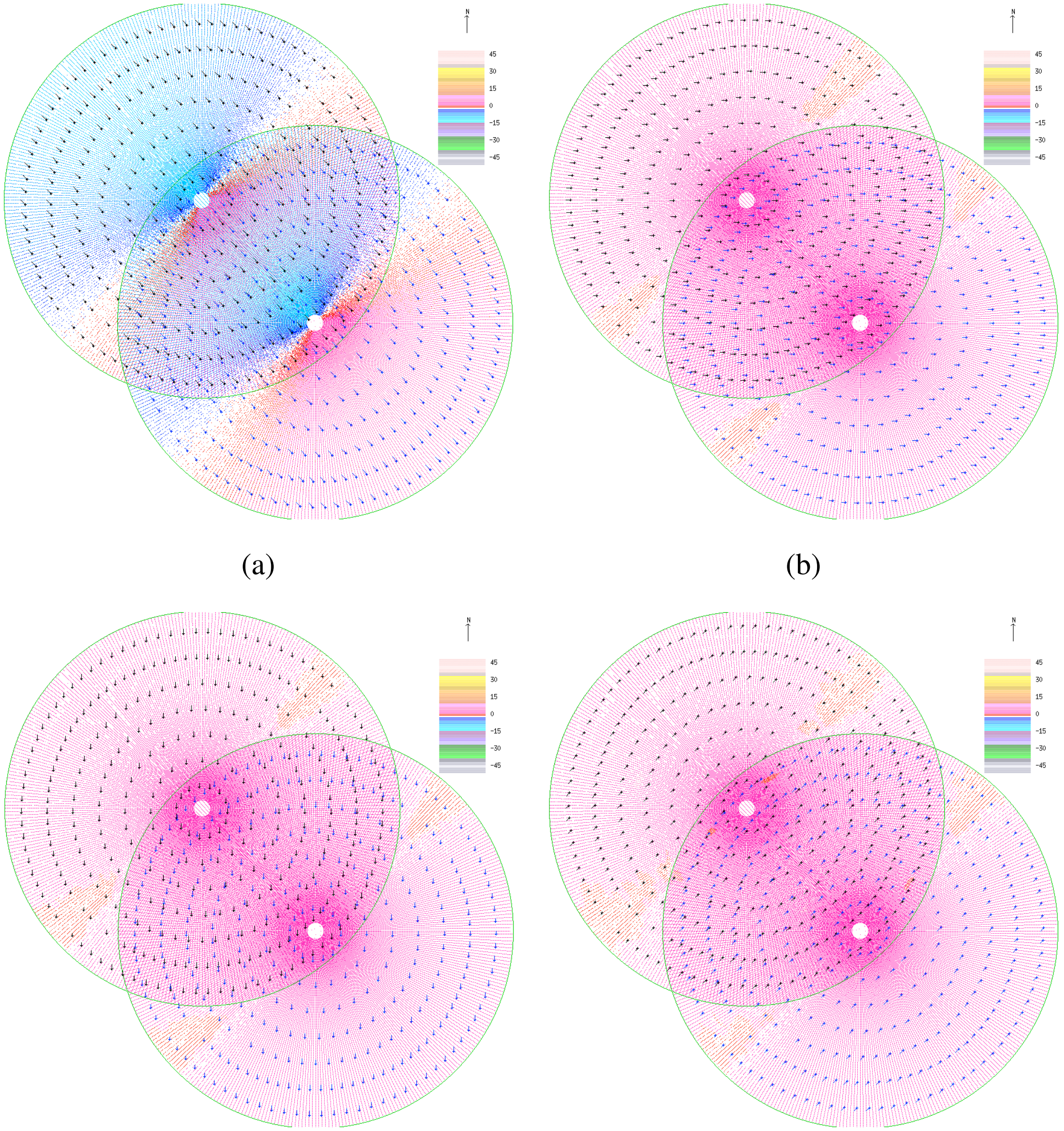
Figure B.15: The correct synthetic velocity of group 3 at variation level $K = 0$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

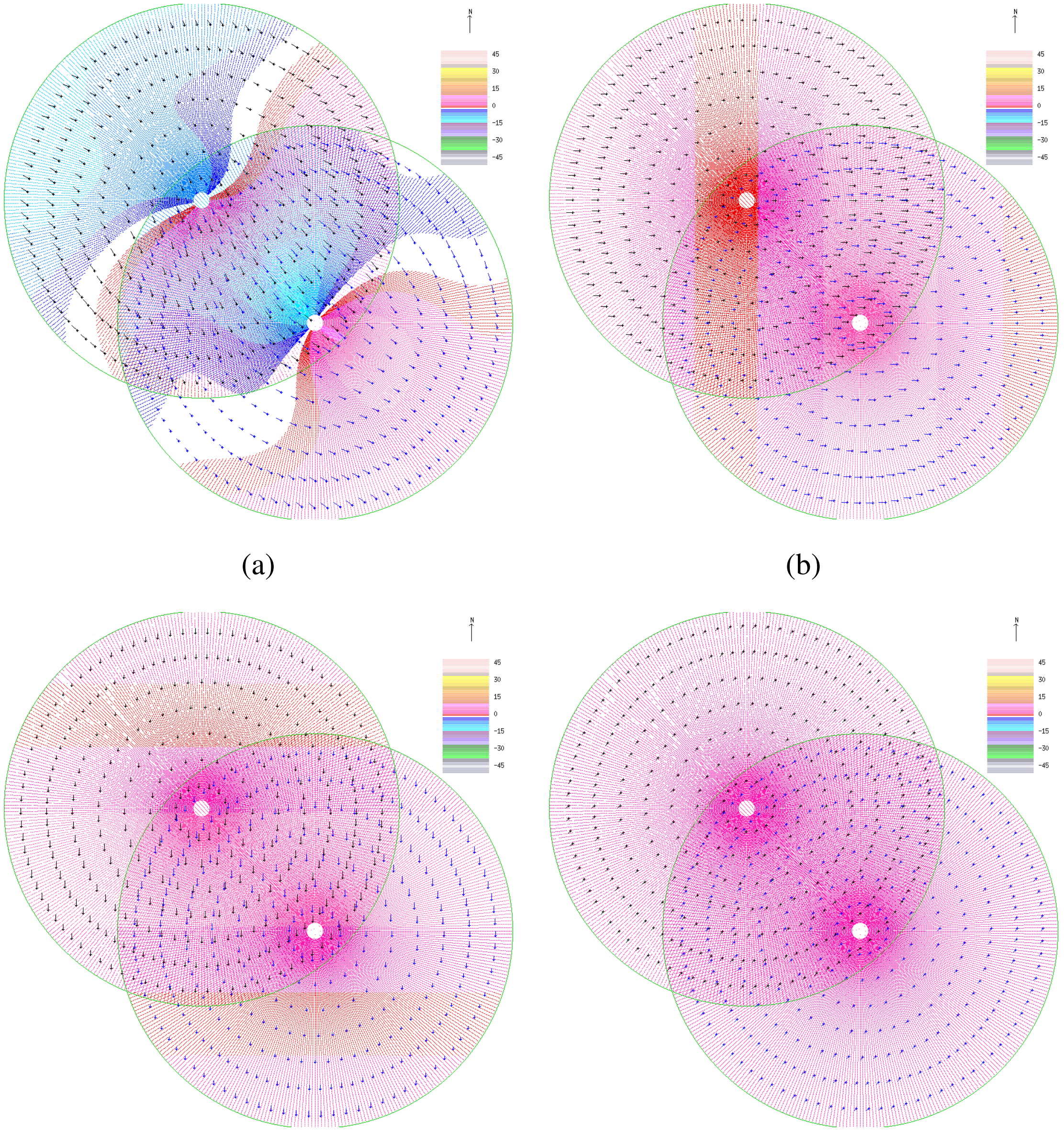
Figure B.16: The single regularization retrieved synthetic velocity of group 3 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

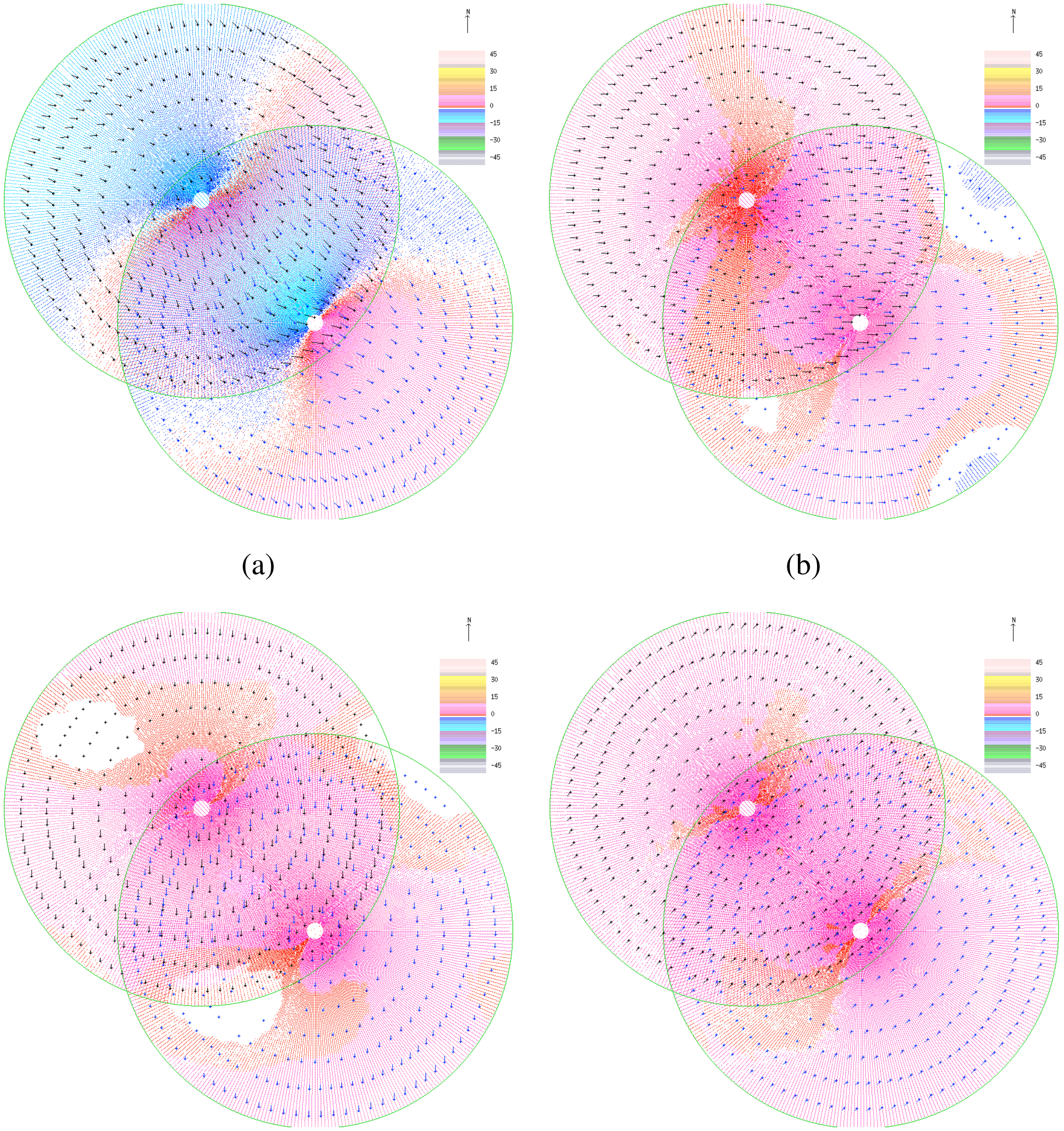
Figure B.17: The dual regularization retrieved synthetic velocity of group 3 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

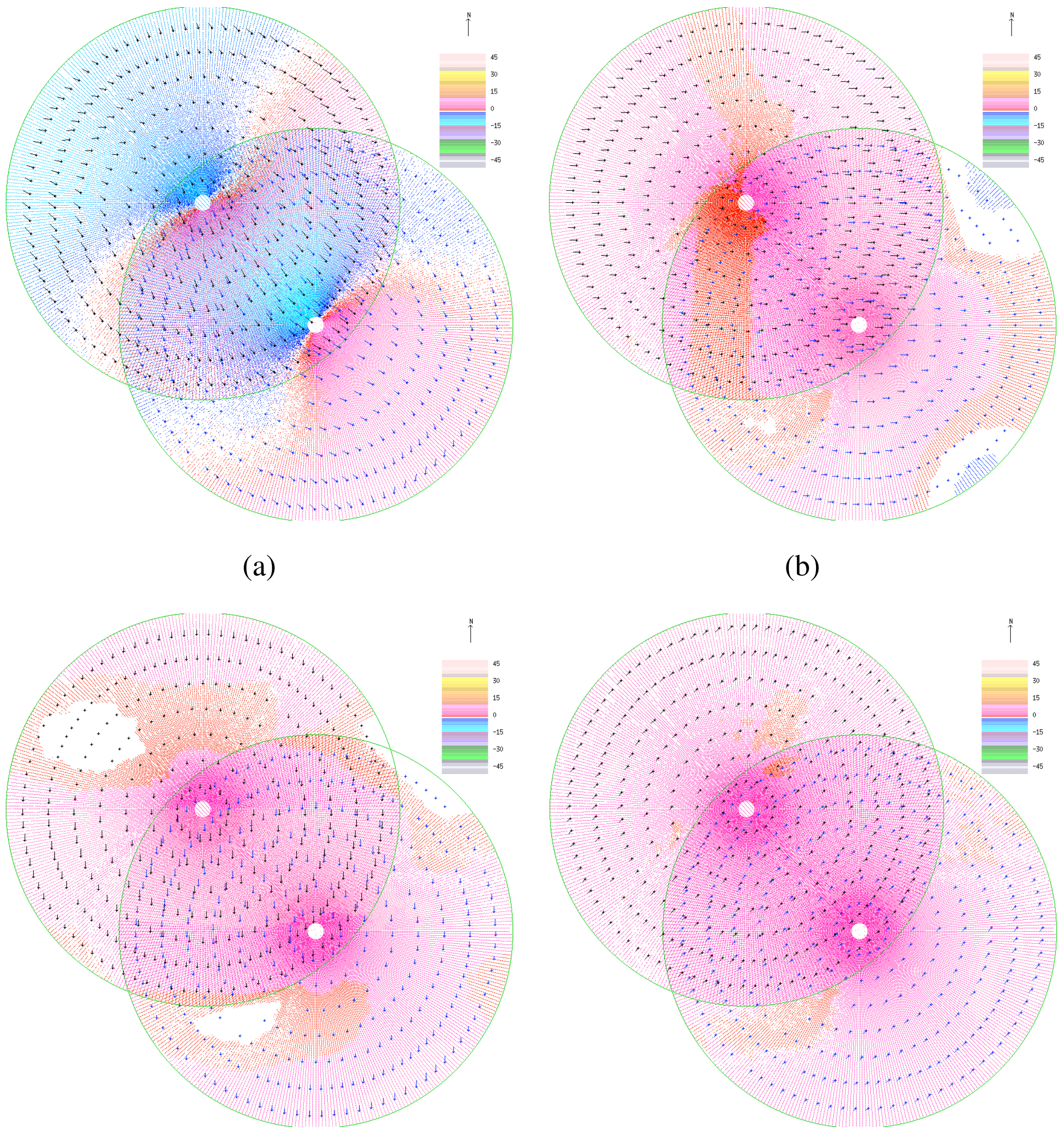
Figure B.18: The correct synthetic velocity of group 3 at variation level $K = 5$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

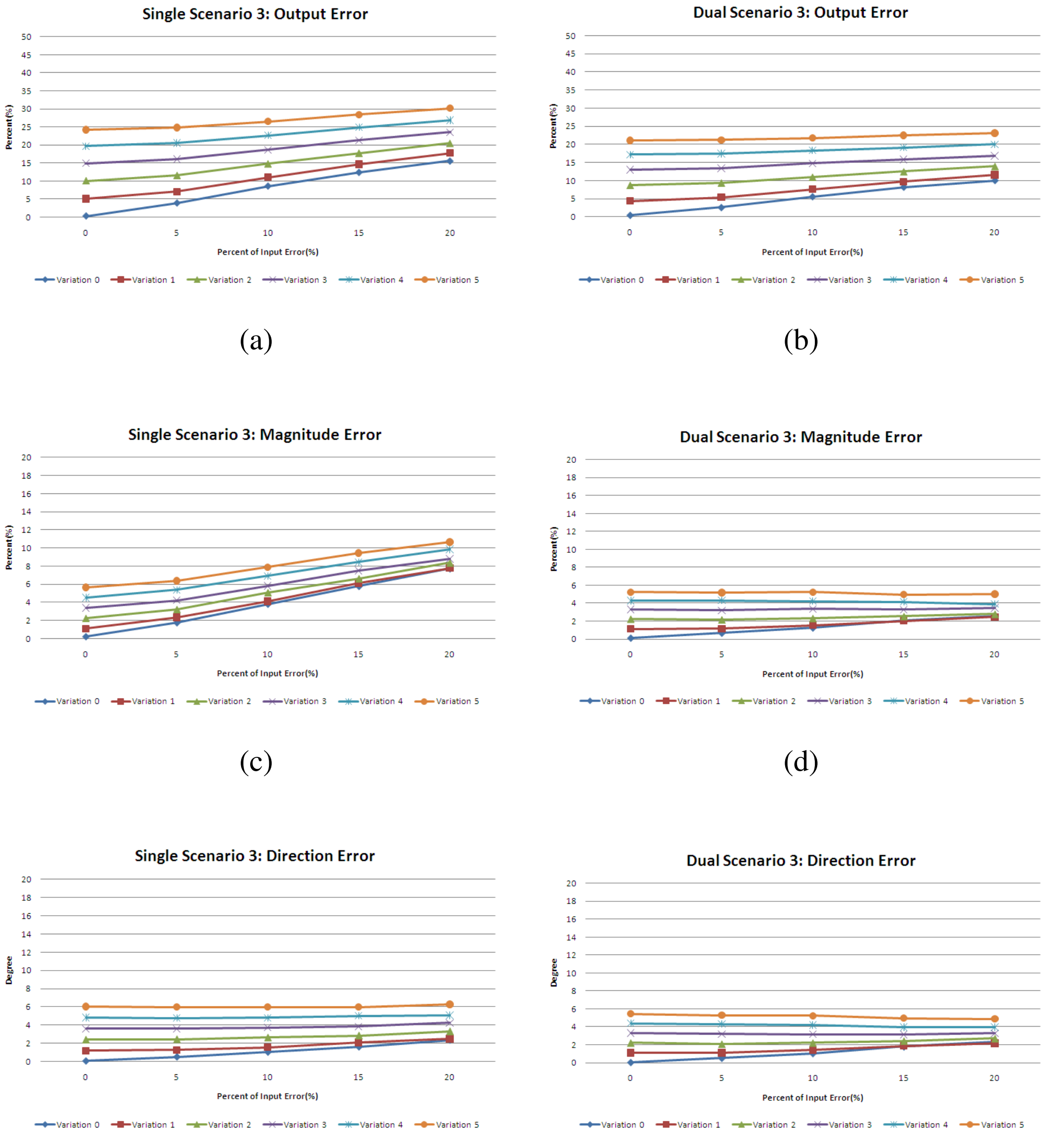
Figure B.19: The single regularization retrieved synthetic velocity of group 3 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

Figure B.20: The dual regularization retrieved synthetic velocity of group 3 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.

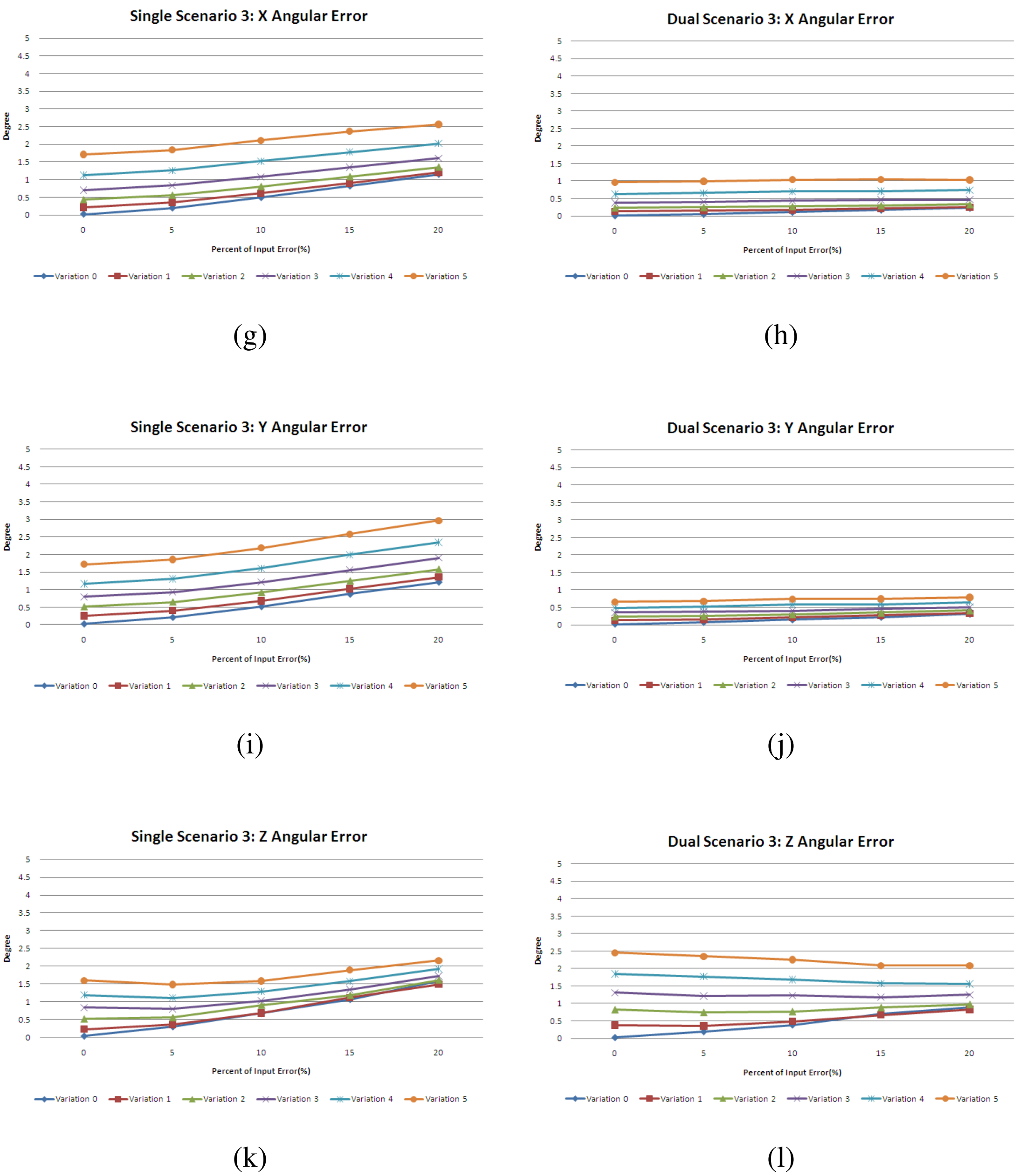


(a) (b)

(c) (d)

(e) (f)

Figure B.21: Result Analysis of Synthetic Data Group 3 using regularization method: (5.0, 5.0, 5.0). The Single Retrieval (a), (c), (e) and Dual Retrieval (b), (d), (f).

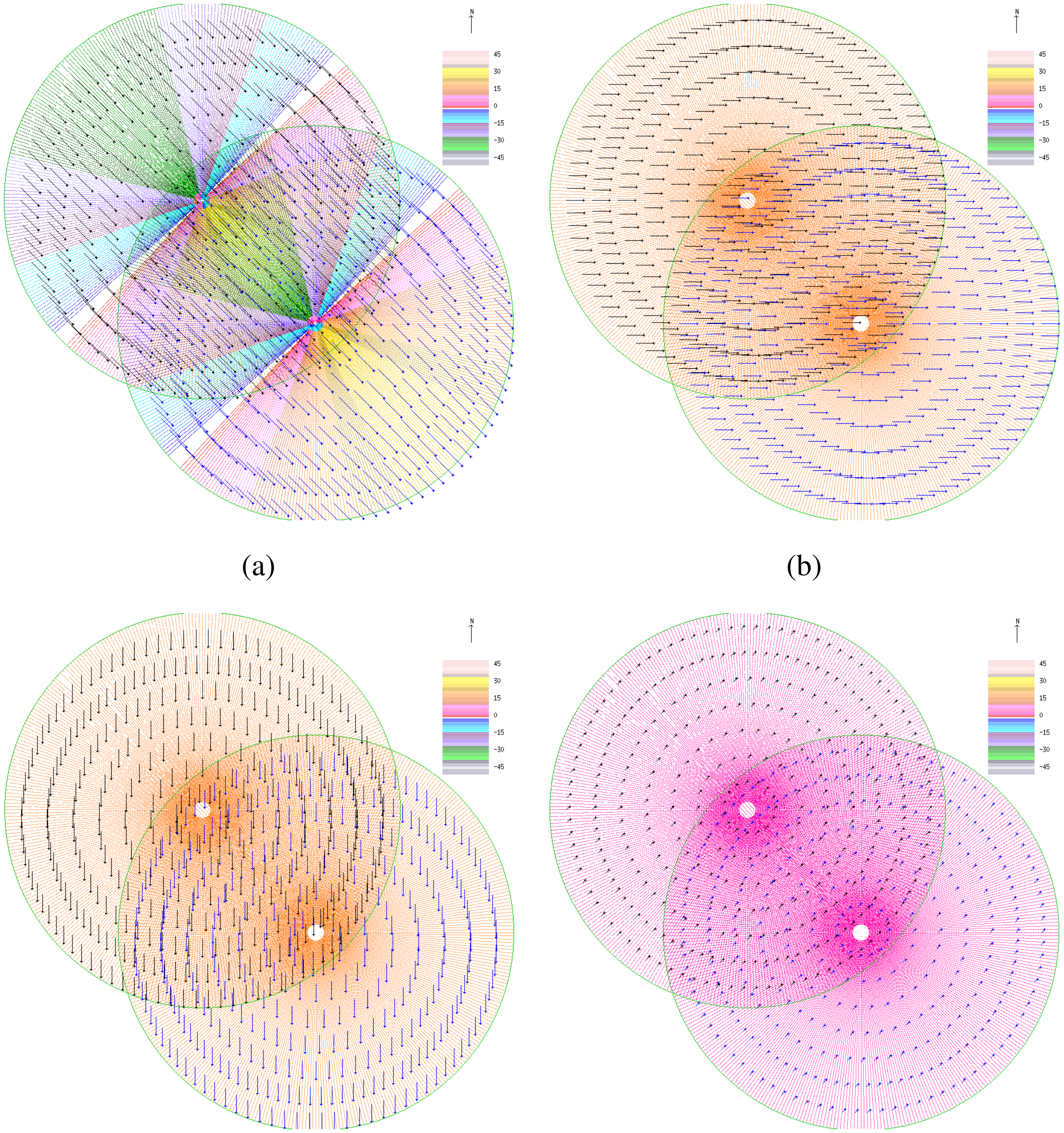


(g) (h)

(i) (j)

(k) (l)

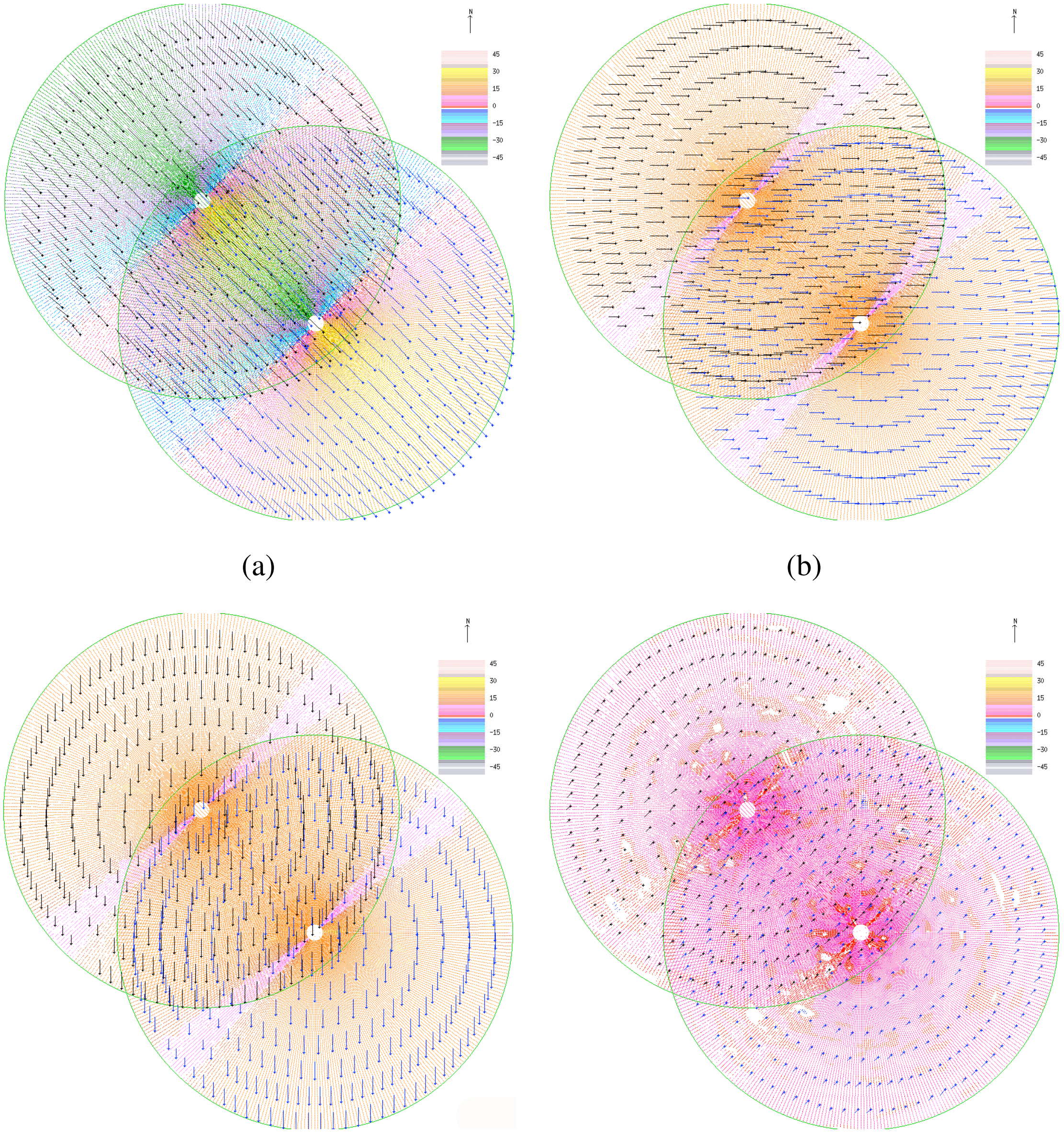
Figure B.21: Result Analysis of Synthetic Data Group 3 using regularization method: (5.0, 5.0, 5.0). The Single Retrieval (g), (i), (k) and Dual retrieval (h), (j), (l).



(a) (b)

(c) (d)

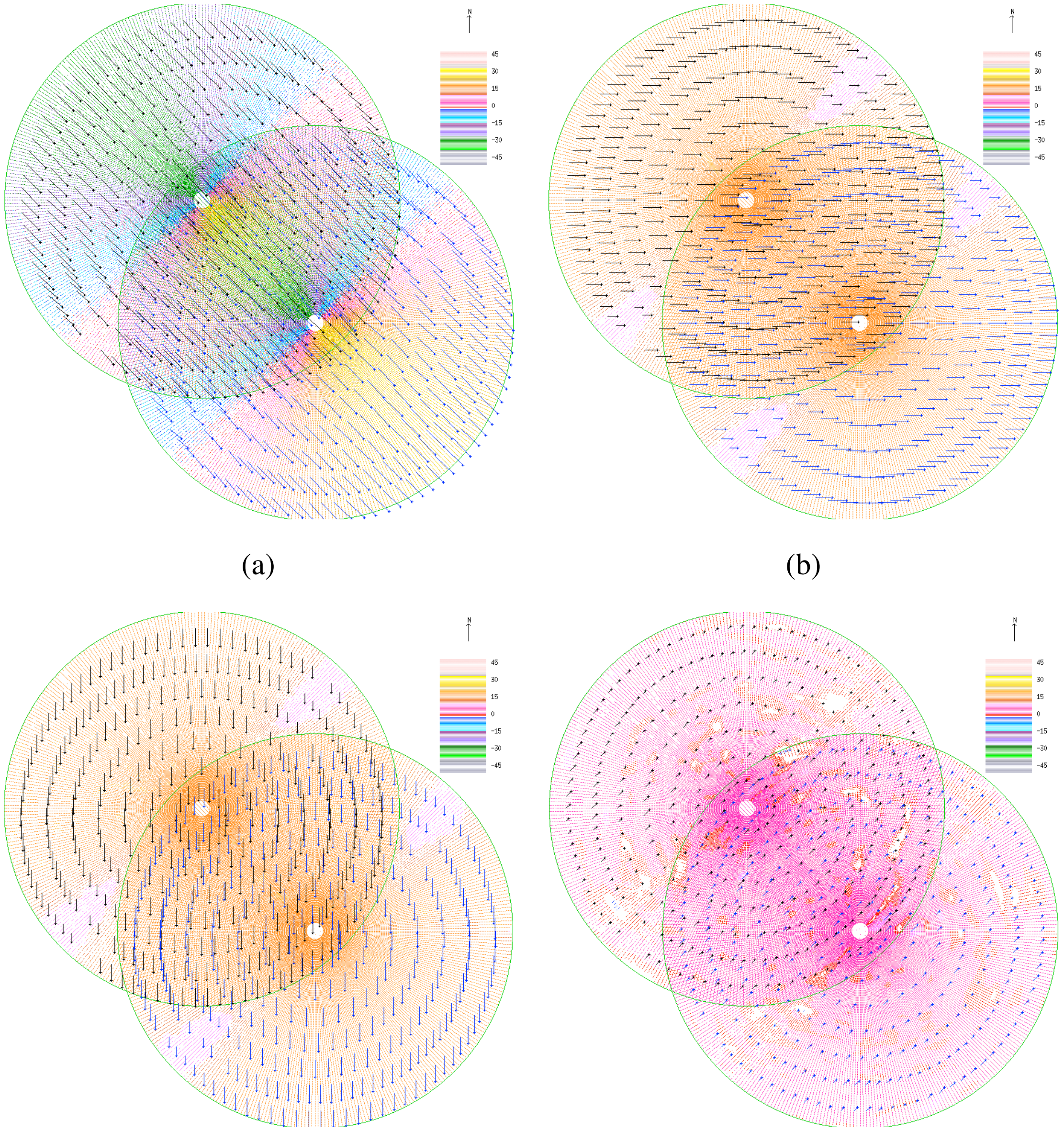
Figure B.22: The correct synthetic velocity of group 4 at variation level $K = 0$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

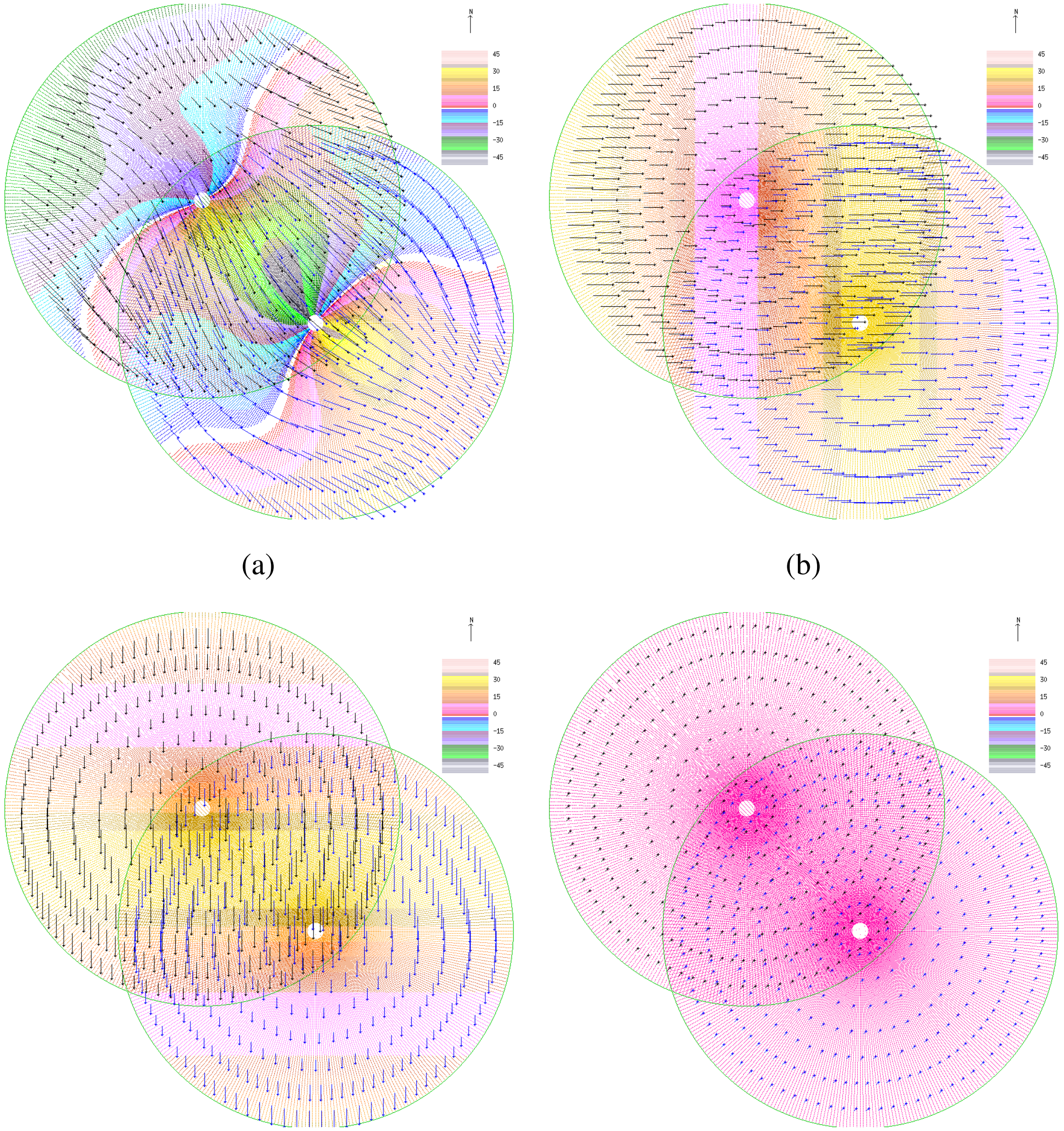
Figure B.23: The single regularization retrieved synthetic velocity of group 4 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

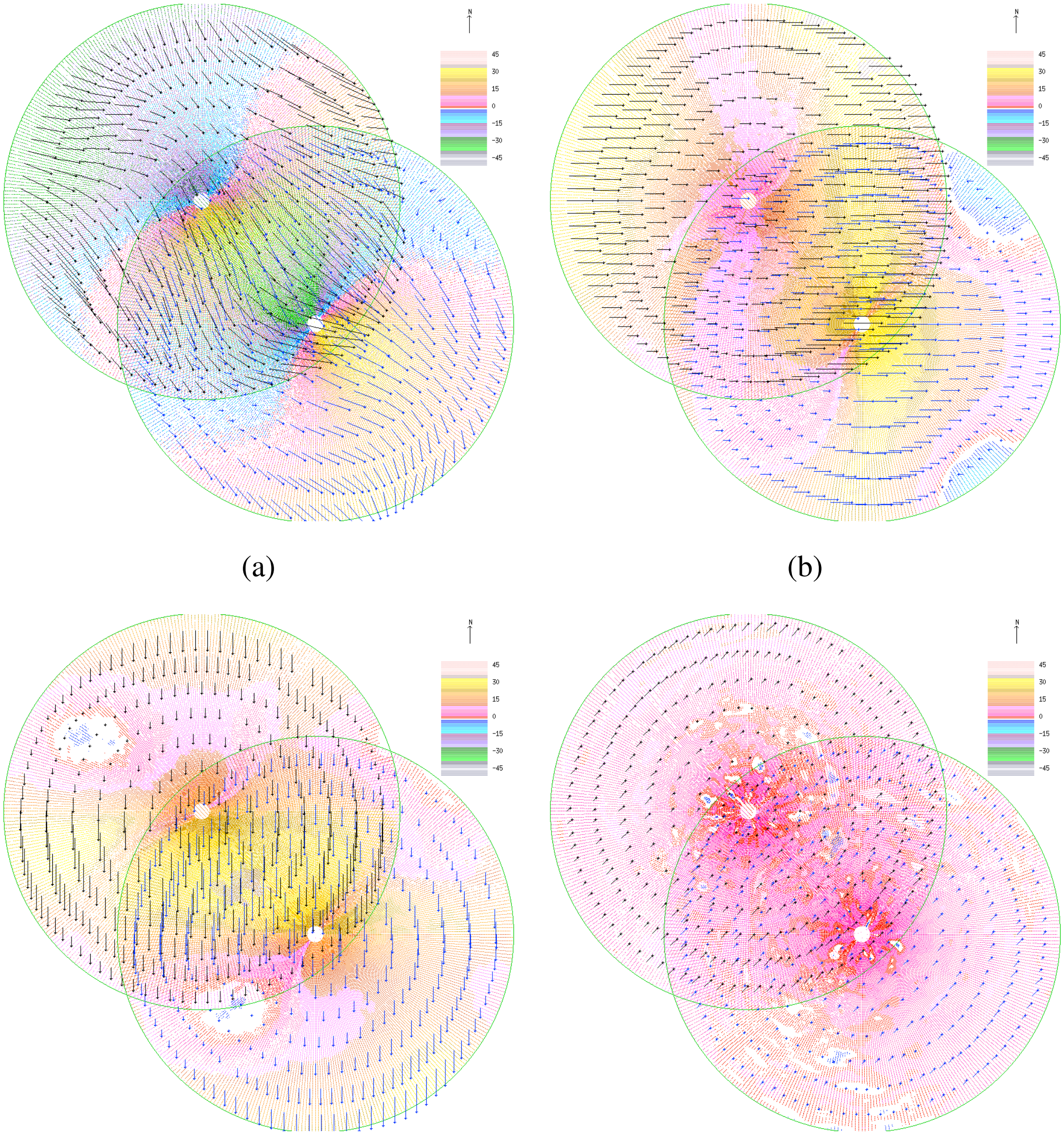
Figure B.24: The dual regularization retrieved synthetic velocity of group 4 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

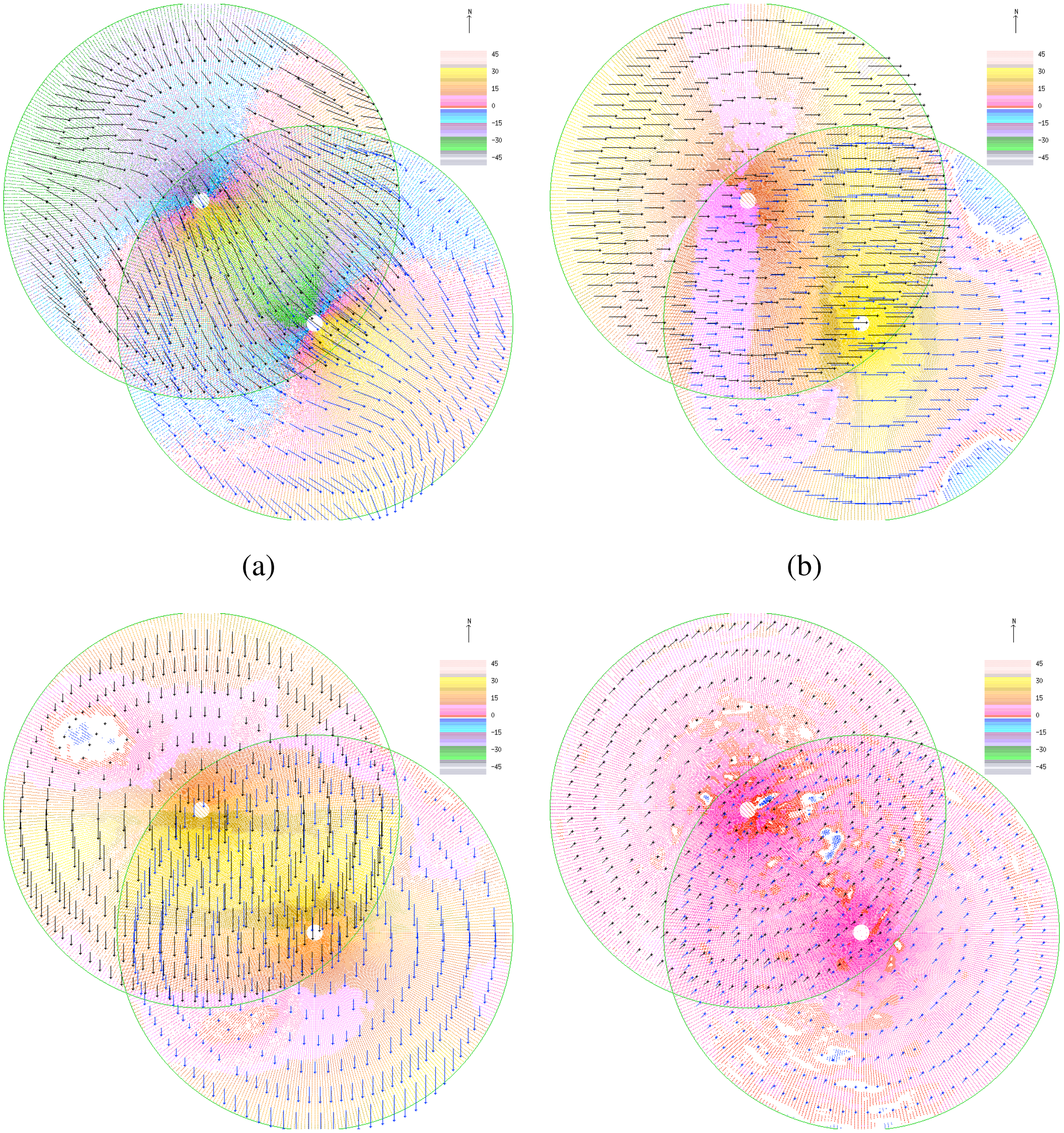
Figure B.25: The correct synthetic velocity of group 4 at variation level $K = 5$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

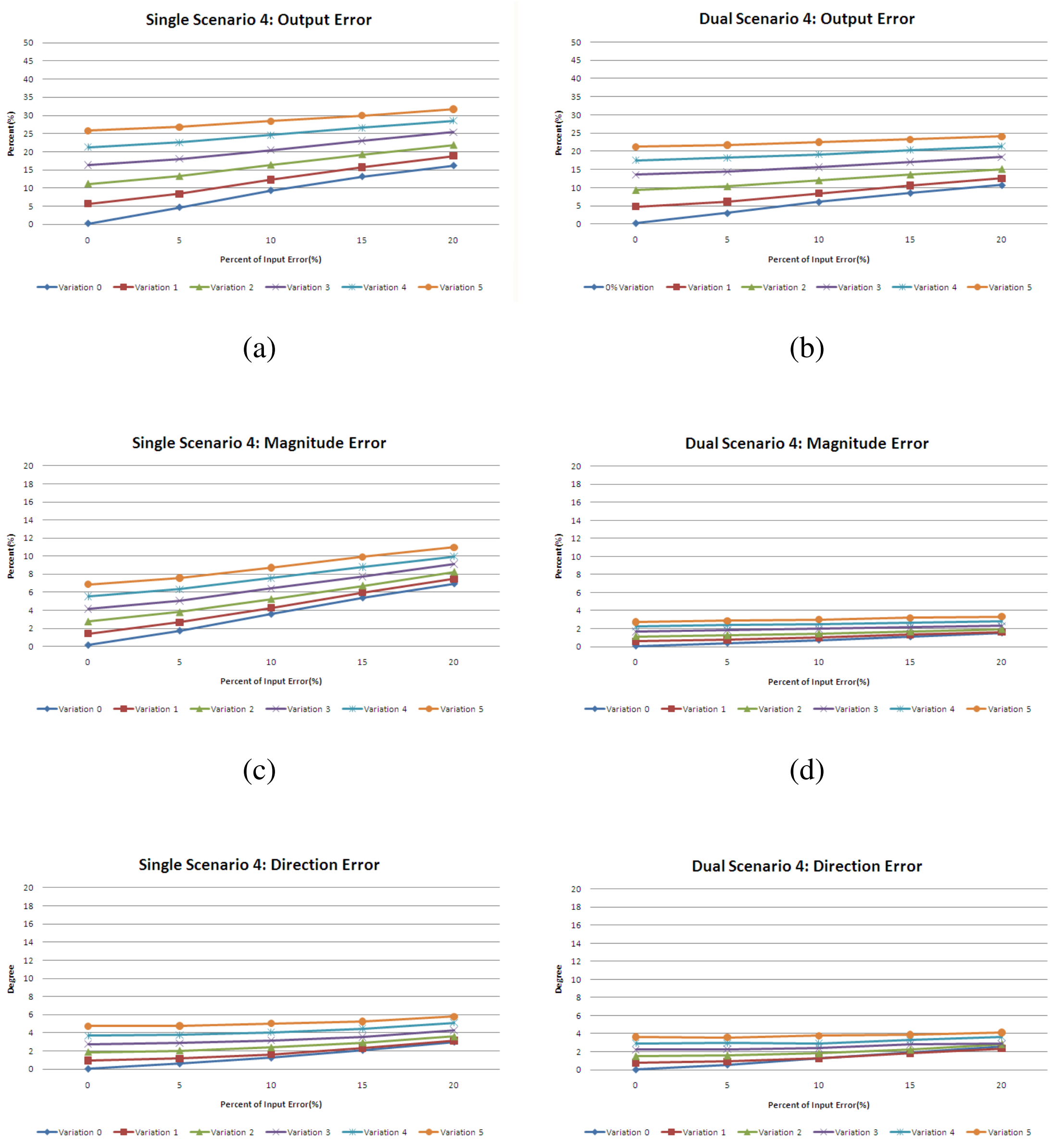
Figure B.26: The single regularization retrieved synthetic velocity of group 4 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

Figure B.27: The dual regularization retrieved synthetic velocity of group 4 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.

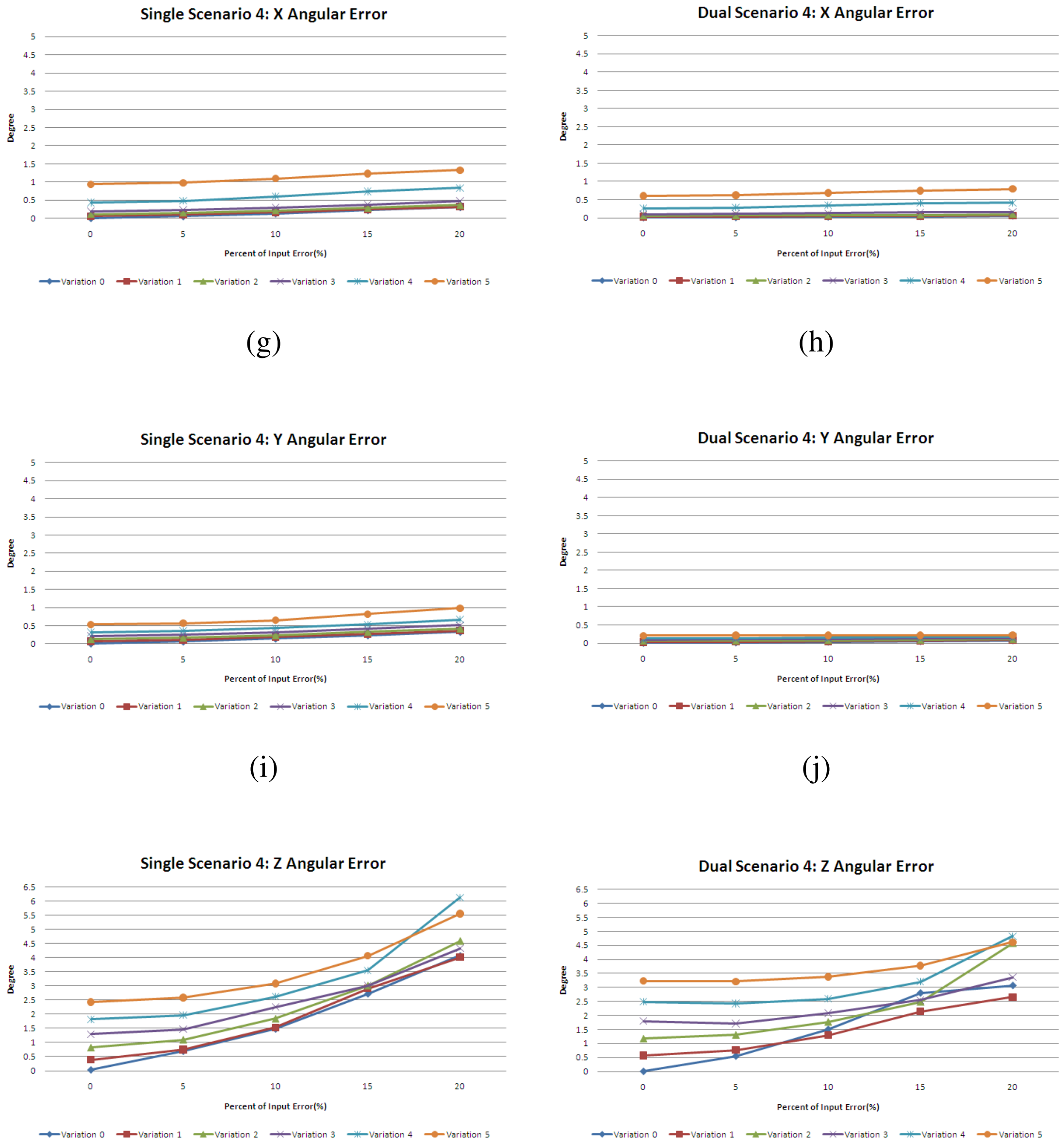


(a) (b)

(c) (d)

(e) (f)

Figure B.28: Result Analysis of Synthetic Data Group 4 using regularization method: (20.0, 20.0, 5.0).The Single Retrieval (a), (c), (e) and Dual Retrieval (b), (d), (f).

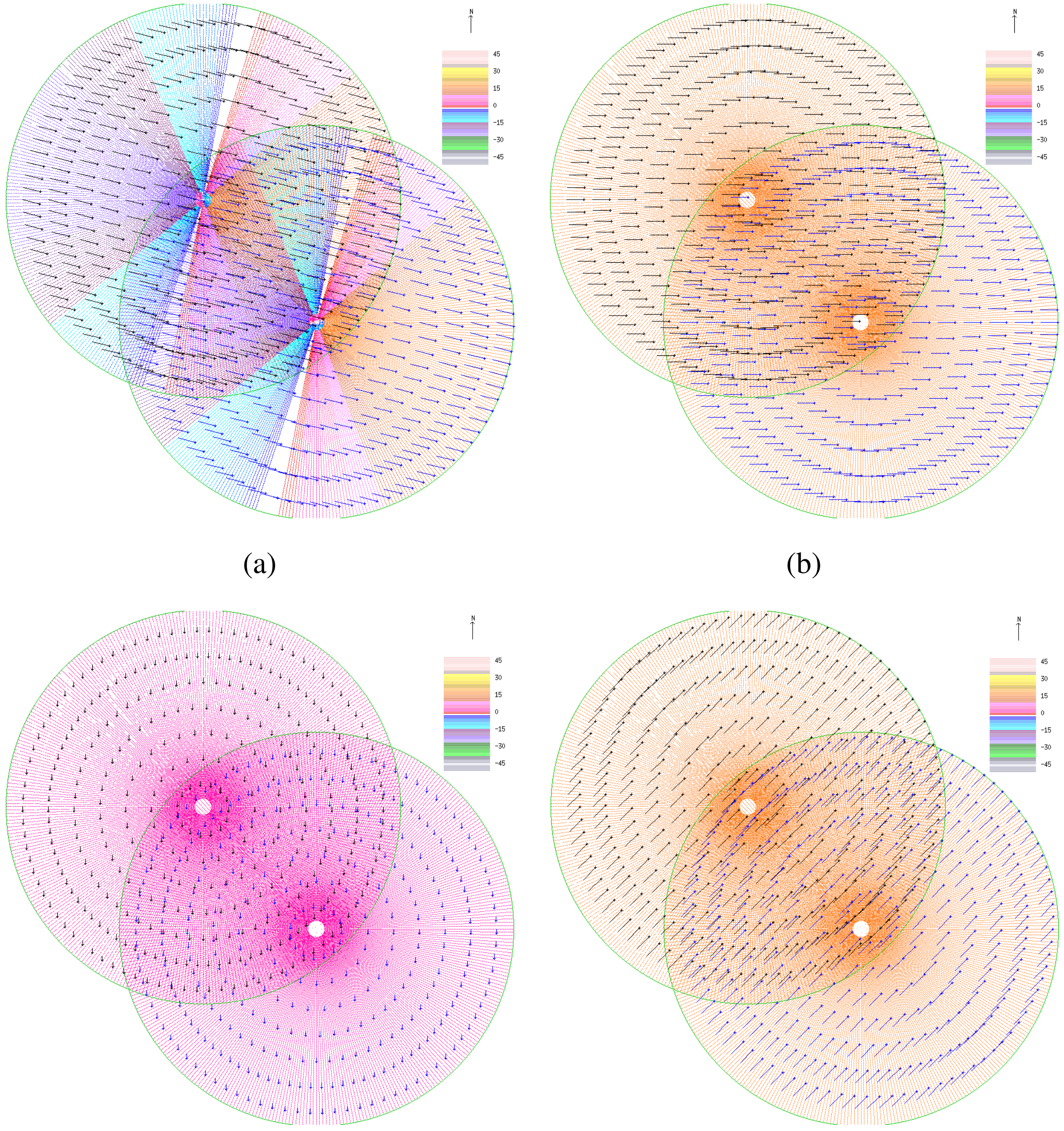


(g) (h)

(i) (j)

(k) (l)

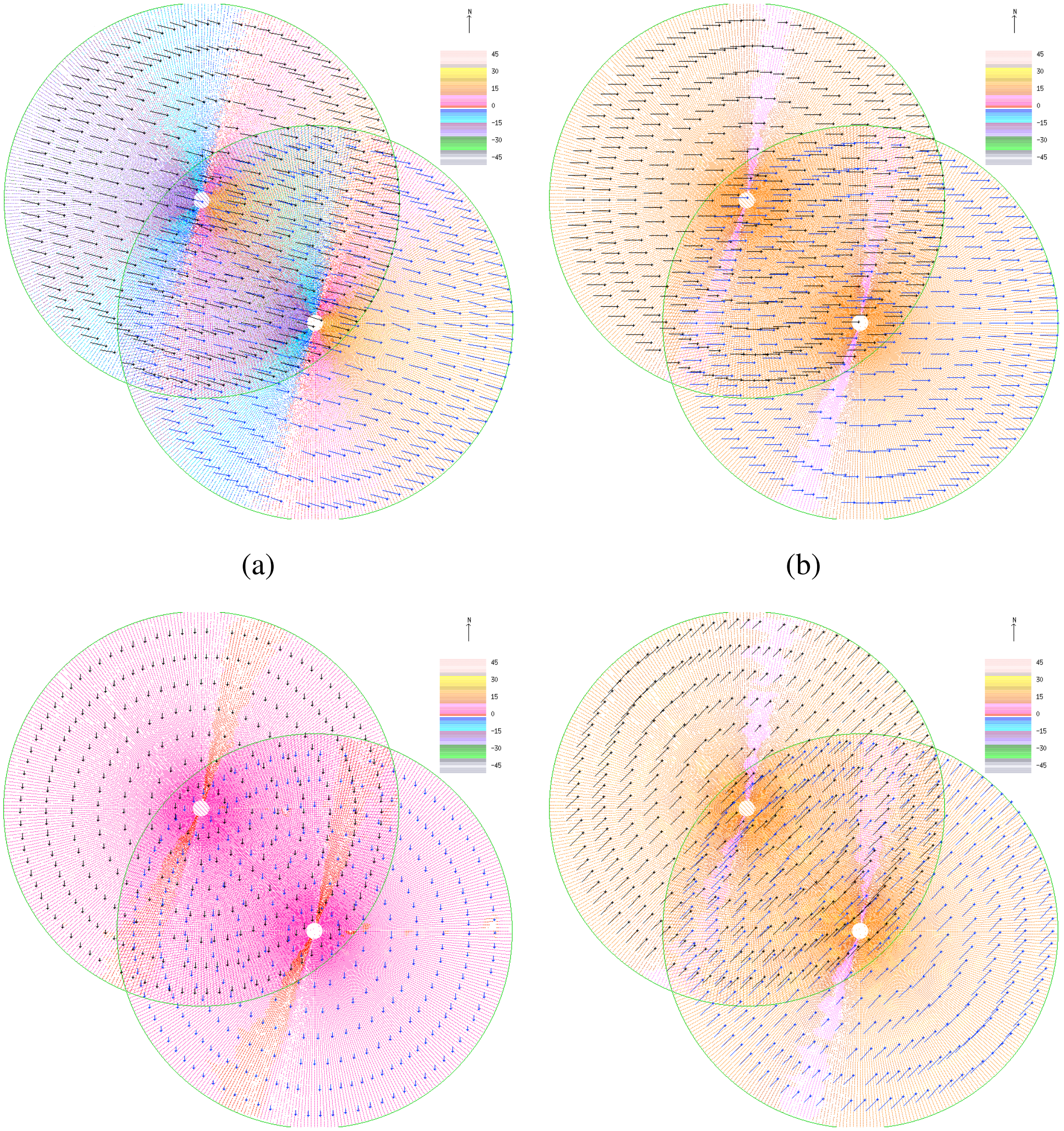
Figure B.28: Result Analysis of Synthetic Data Group 4 using regularization method: (20.0, 20.0, 5.0). The Single Retrieval (g), (i), (k) and Dual retrieval (h), (j), (l).



(a) (b)

(c) (d)

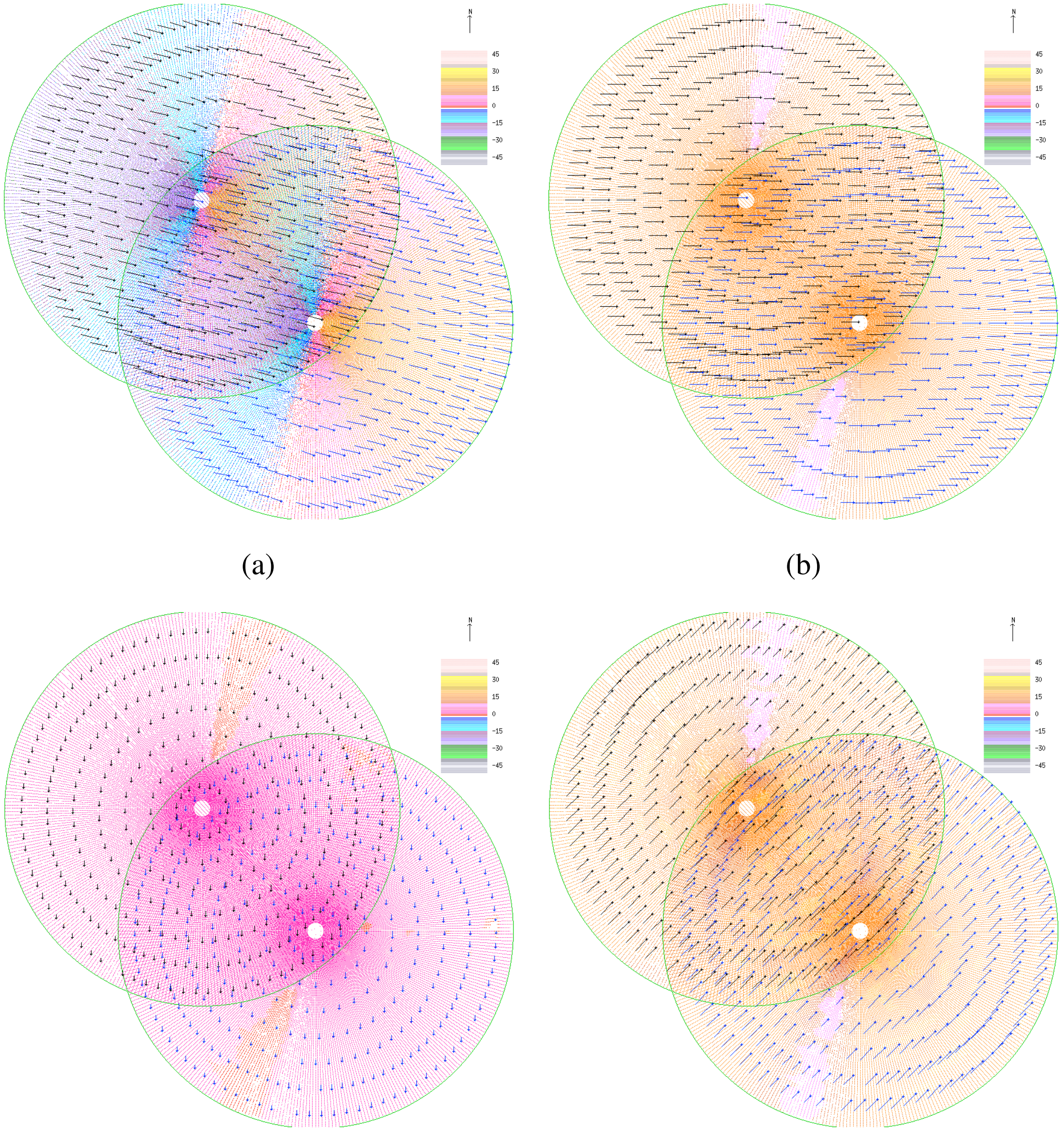
Figure B.29: The correct synthetic velocity of group 5 at variation level $K = 0$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

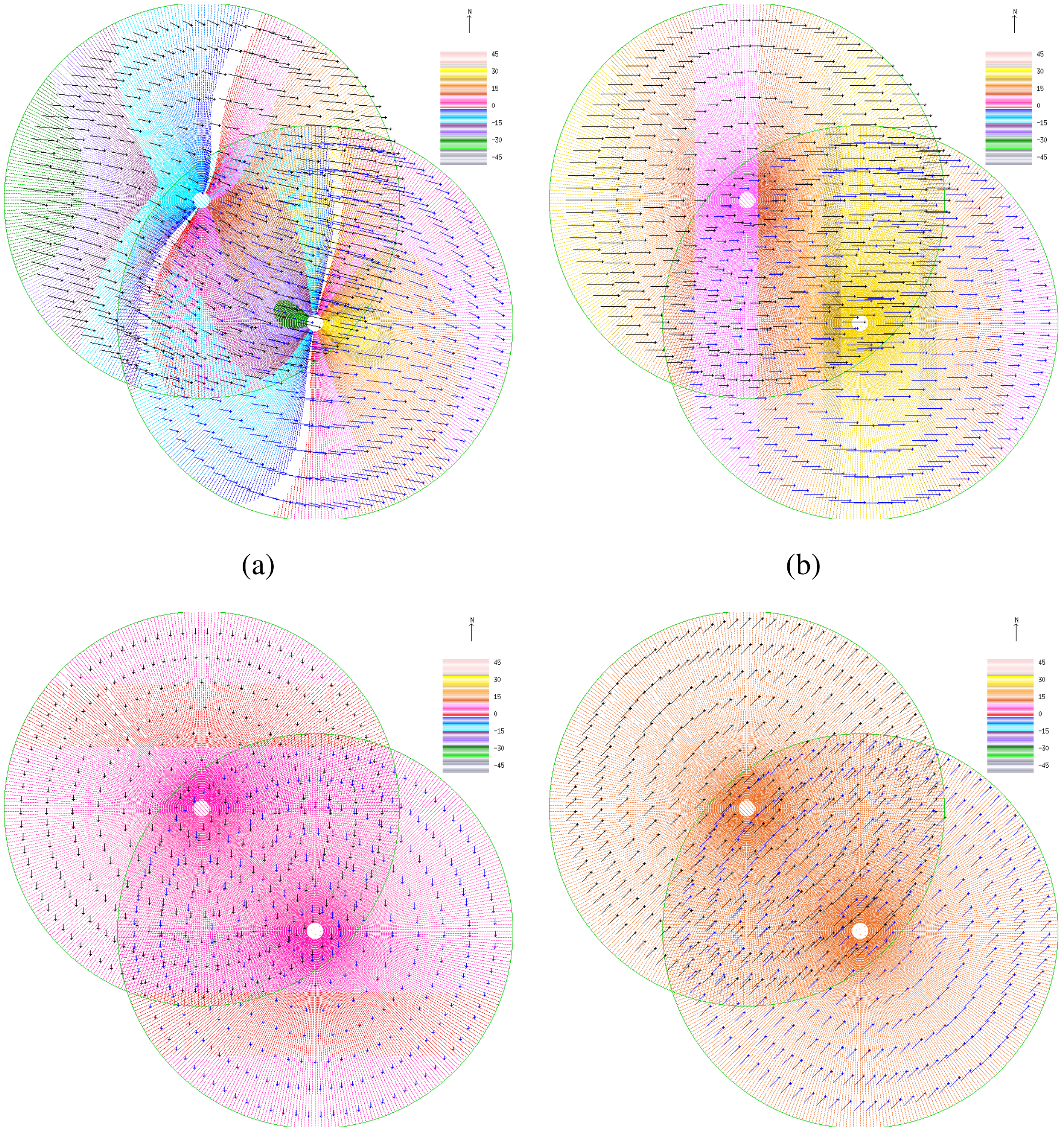
Figure B.30: The single regularization retrieved synthetic velocity of group 5 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

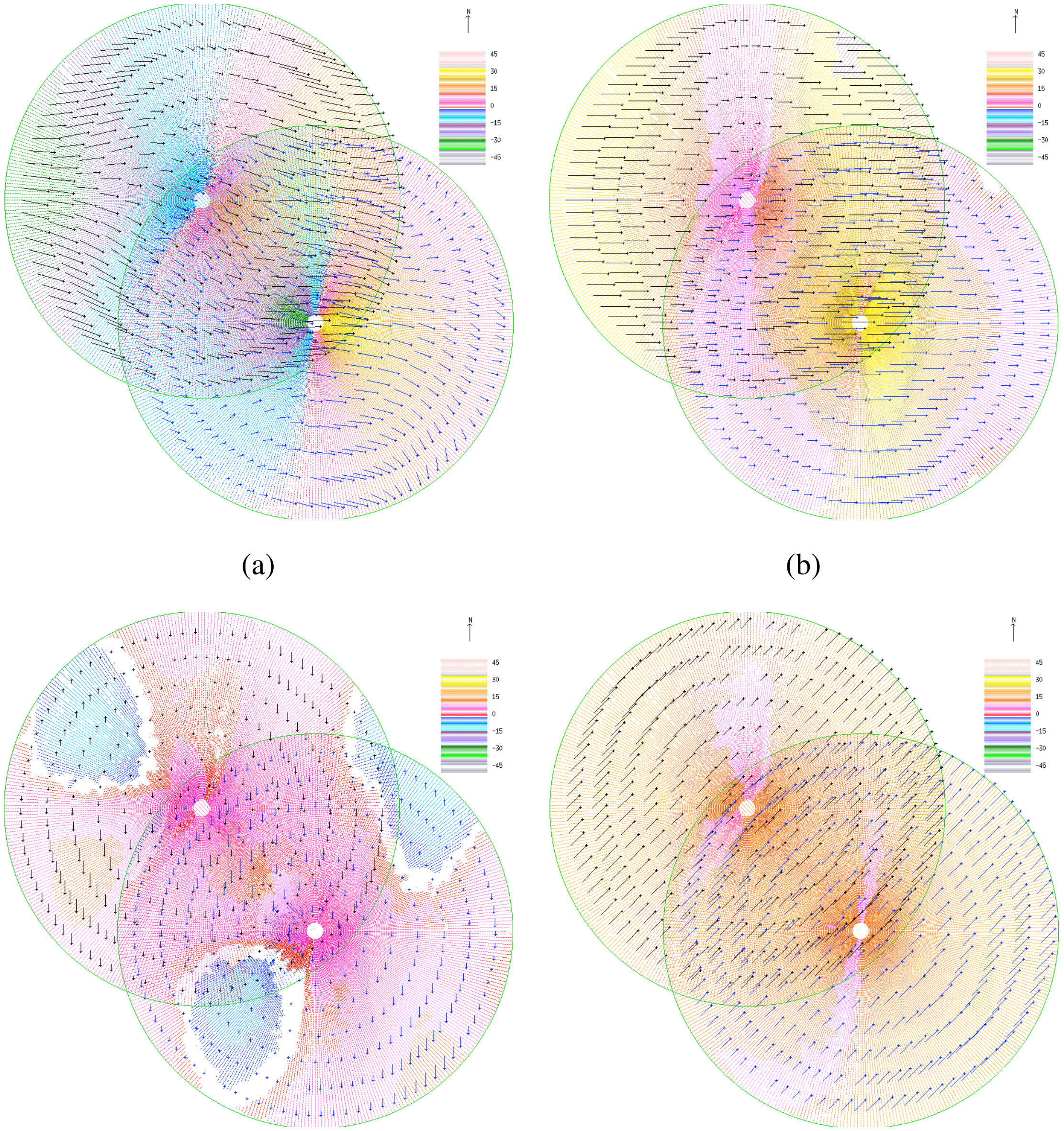
Figure B.31: The dual regularization retrieved synthetic velocity of group 5 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

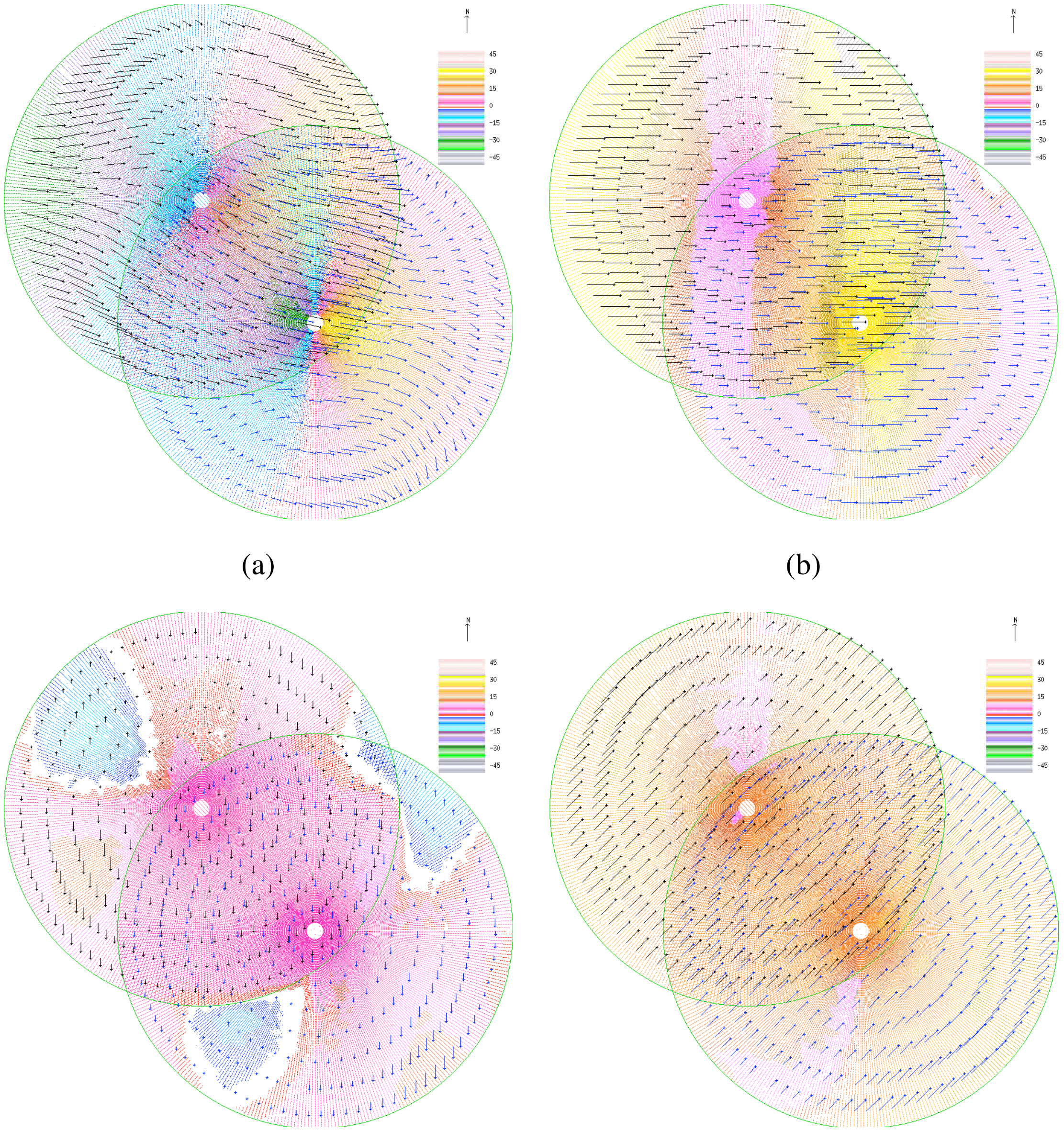
Figure B.32: The correct synthetic velocity of group 5 at variation level $K = 5$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

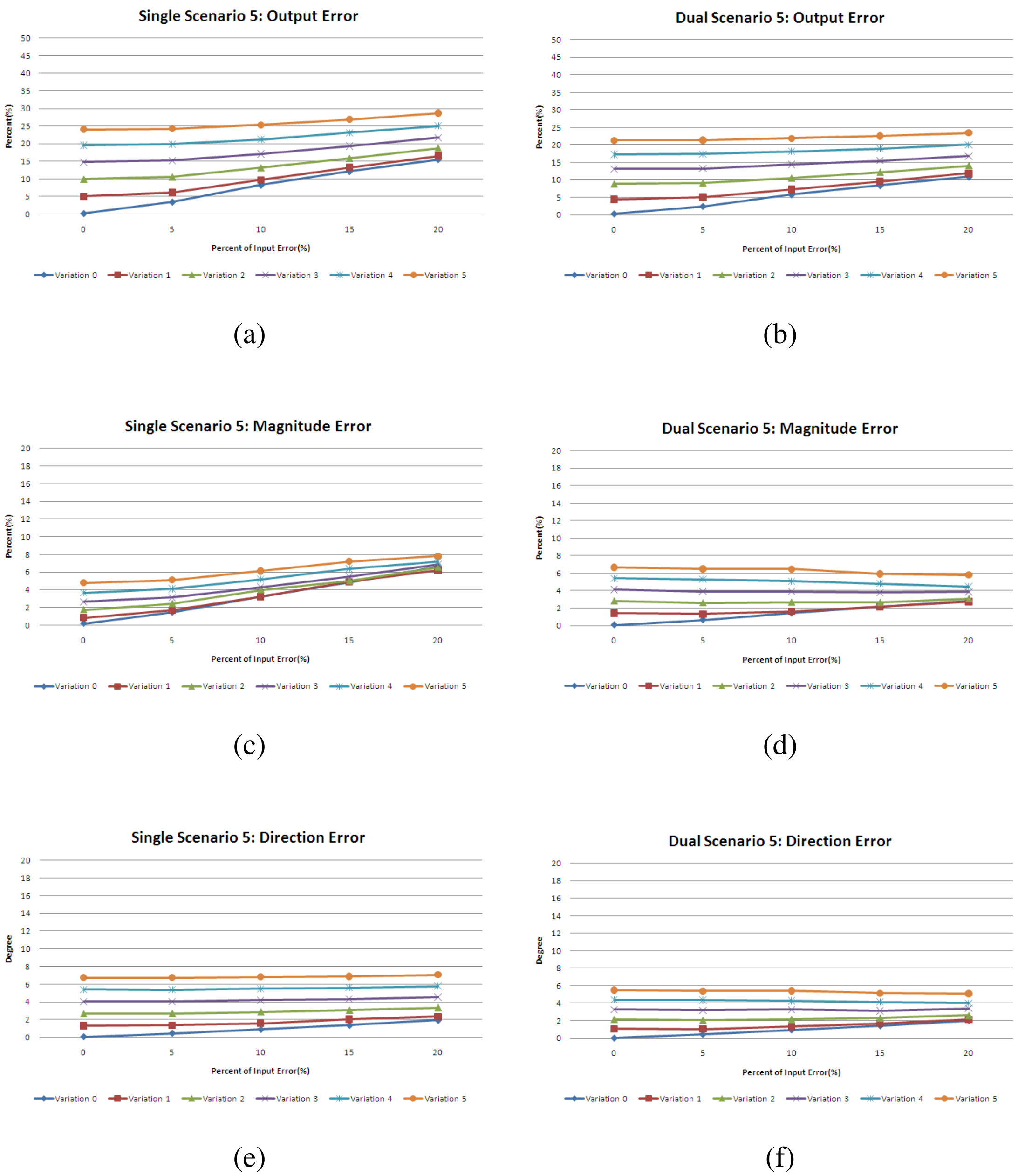
Figure B.33: The single regularization retrieved synthetic velocity of group 5 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

Figure B.34: The dual regularization retrieved synthetic velocity of group 5 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



Single Scenario 5: Output Error

Dual Scenario 5: Output Error

(a)

(b)

Single Scenario 5: Magnitude Error

Dual Scenario 5: Magnitude Error

(c)

(d)

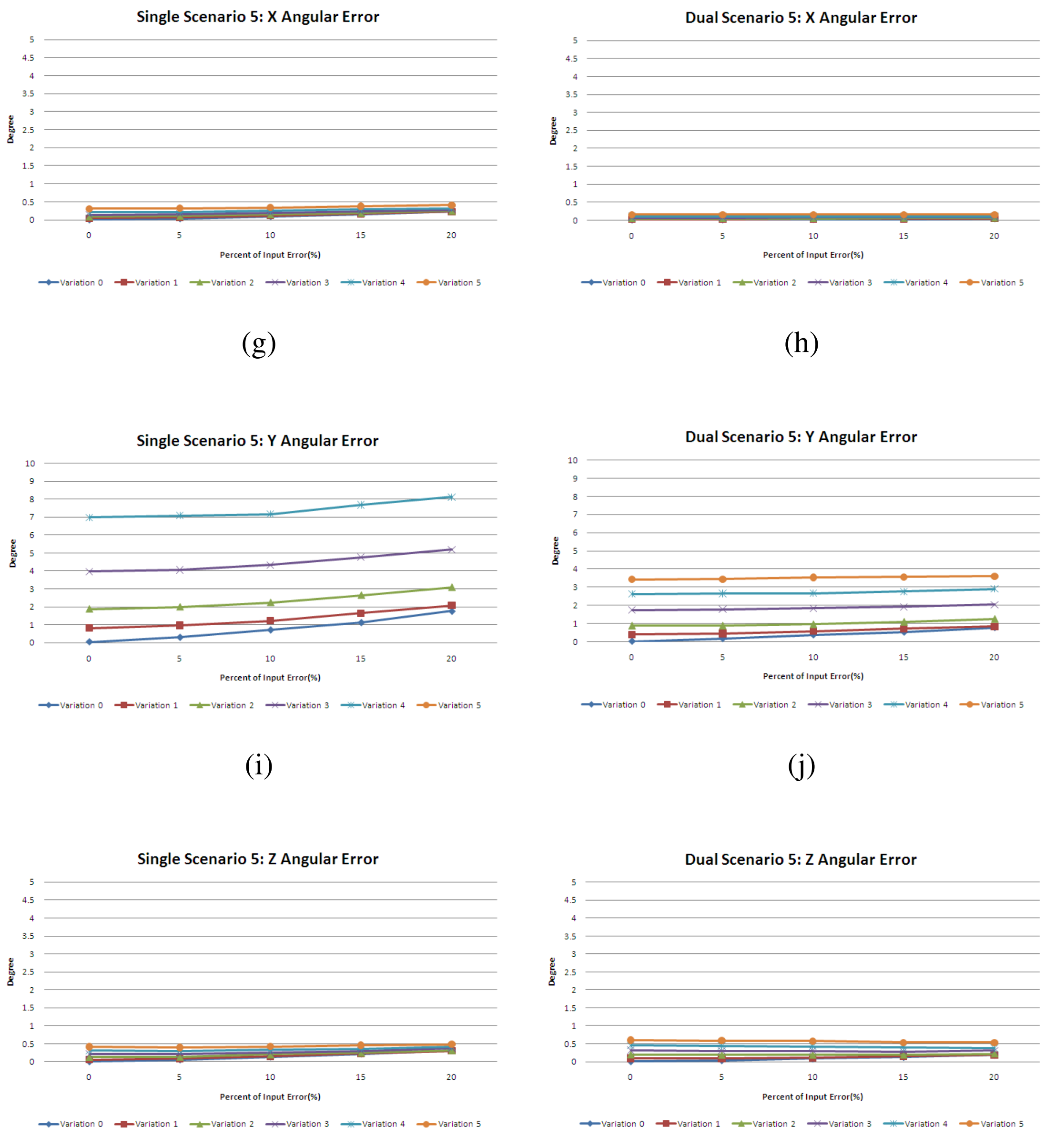
Single Scenario 5: Direction Error

Dual Scenario 5: Direction Error

(e)

(f)

Figure B.35: Result Analysis of Synthetic Data Group 5 using regularization method: (20.0, 5.0, 20.0). The Single Retrieval (a), (c), (e) and Dual Retrieval (b), (d), (f).

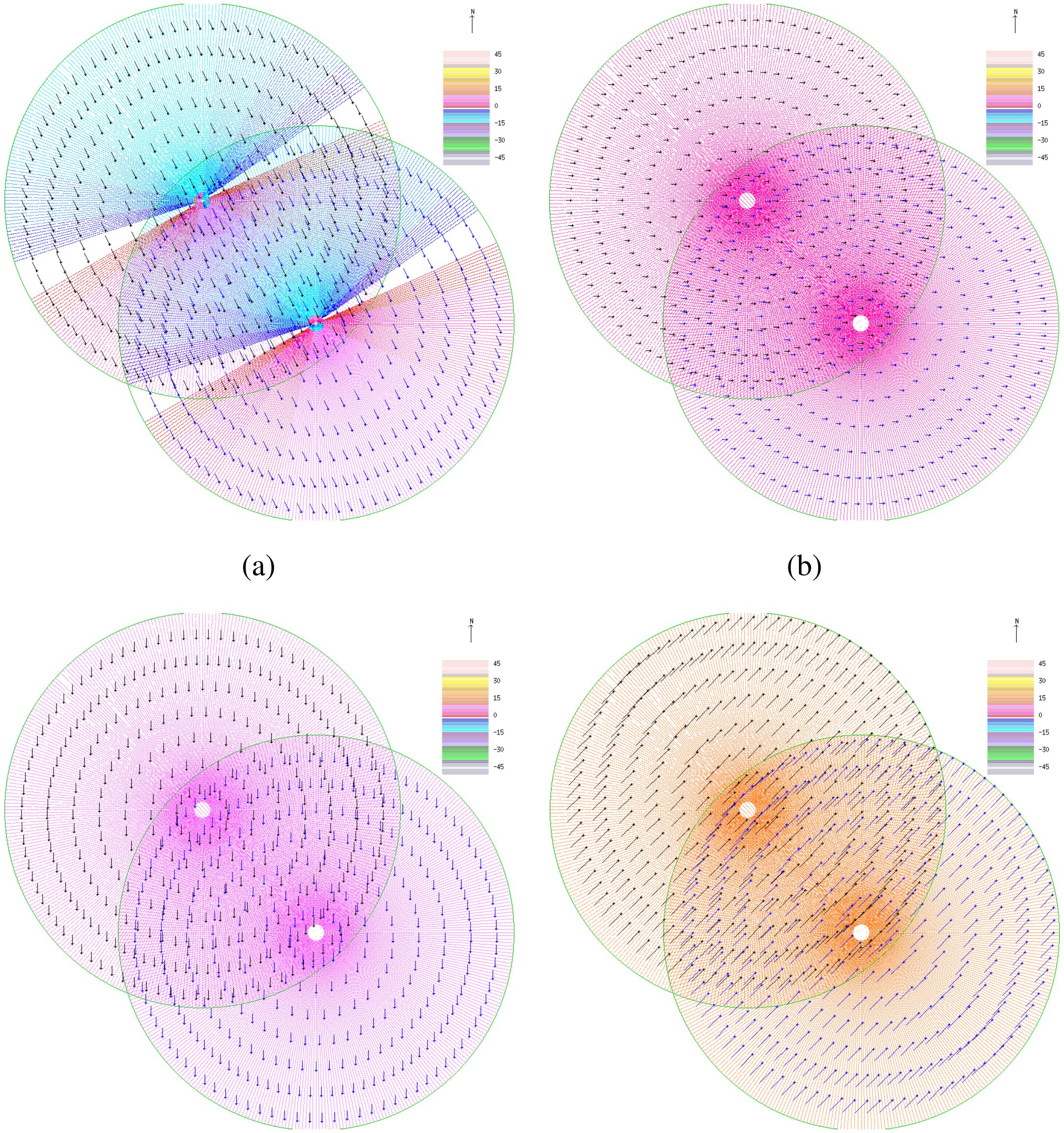


(g) (h)

(i) (j)

(k) (l)

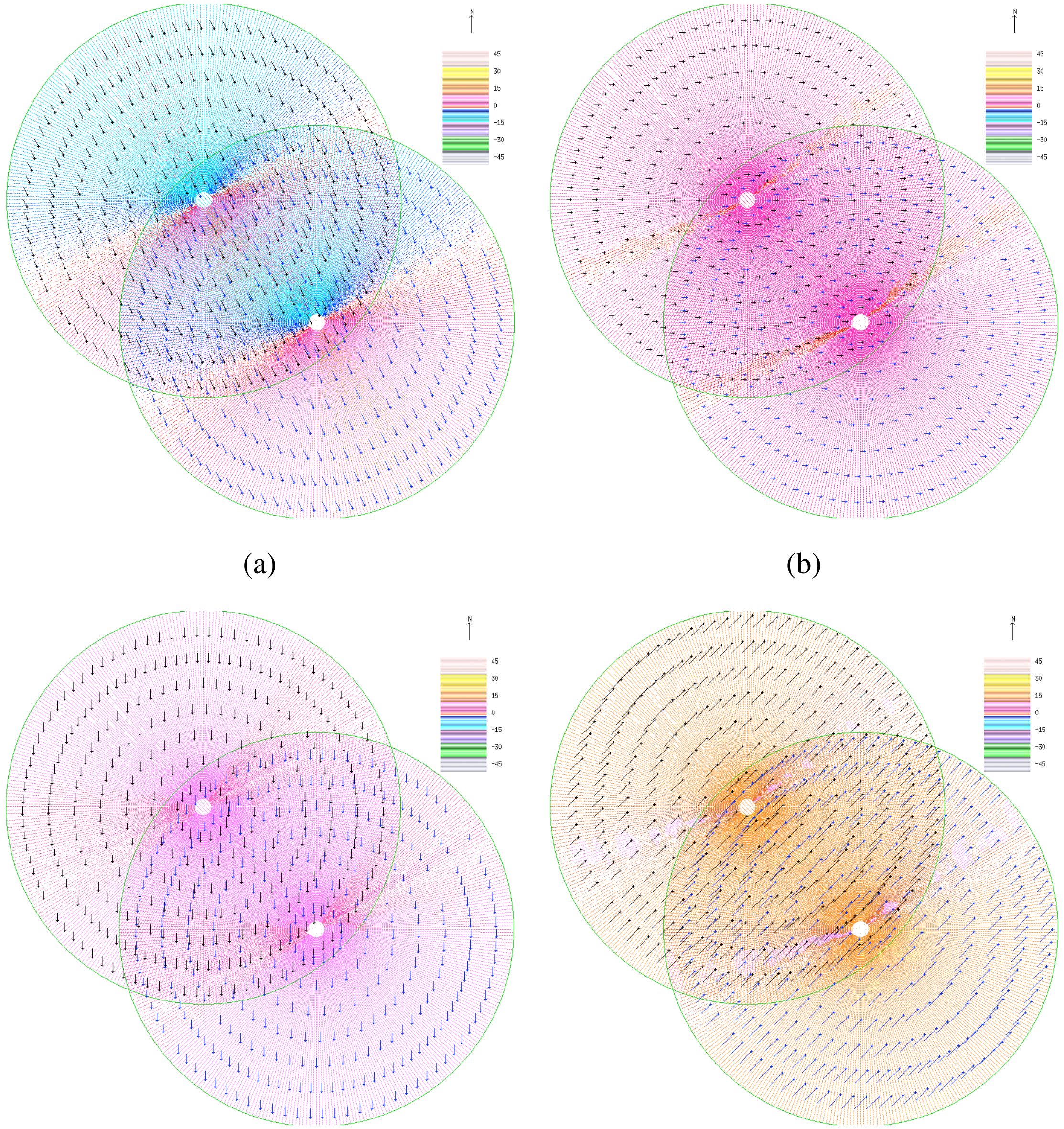
Figure B.35: Result Analysis of Synthetic Data Group 5 using regularization method: (20.0, 5.0, 20.0). The Single Retrieval (g), (i), (k) and Dual retrieval (h), (j), (l).



(a) (b)

(c) (d)

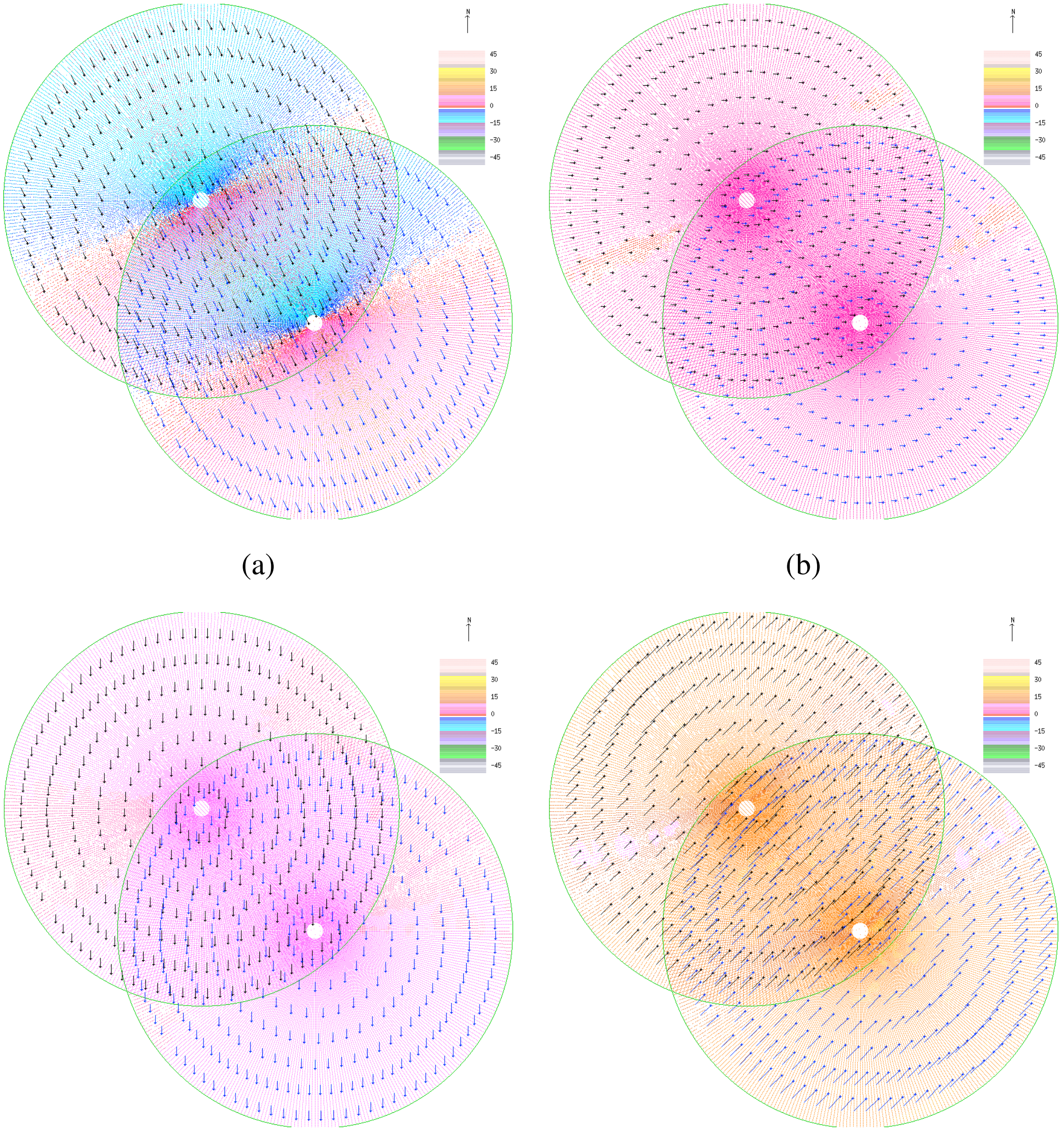
Figure B.36: The correct synthetic velocity of group 2 at variation level $K = 0$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

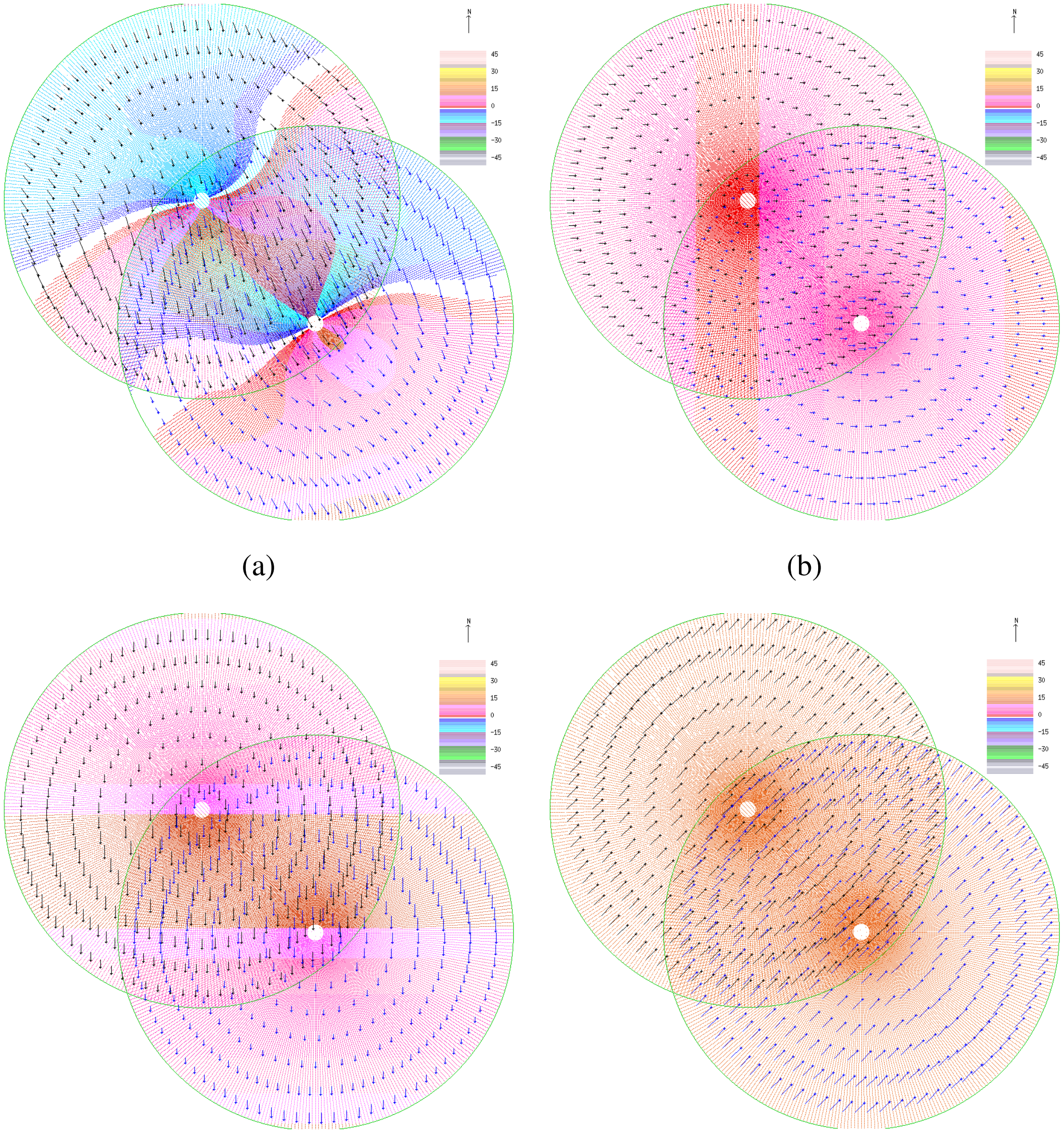
Figure B.37: The single regularization retrieved synthetic velocity of group 2 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

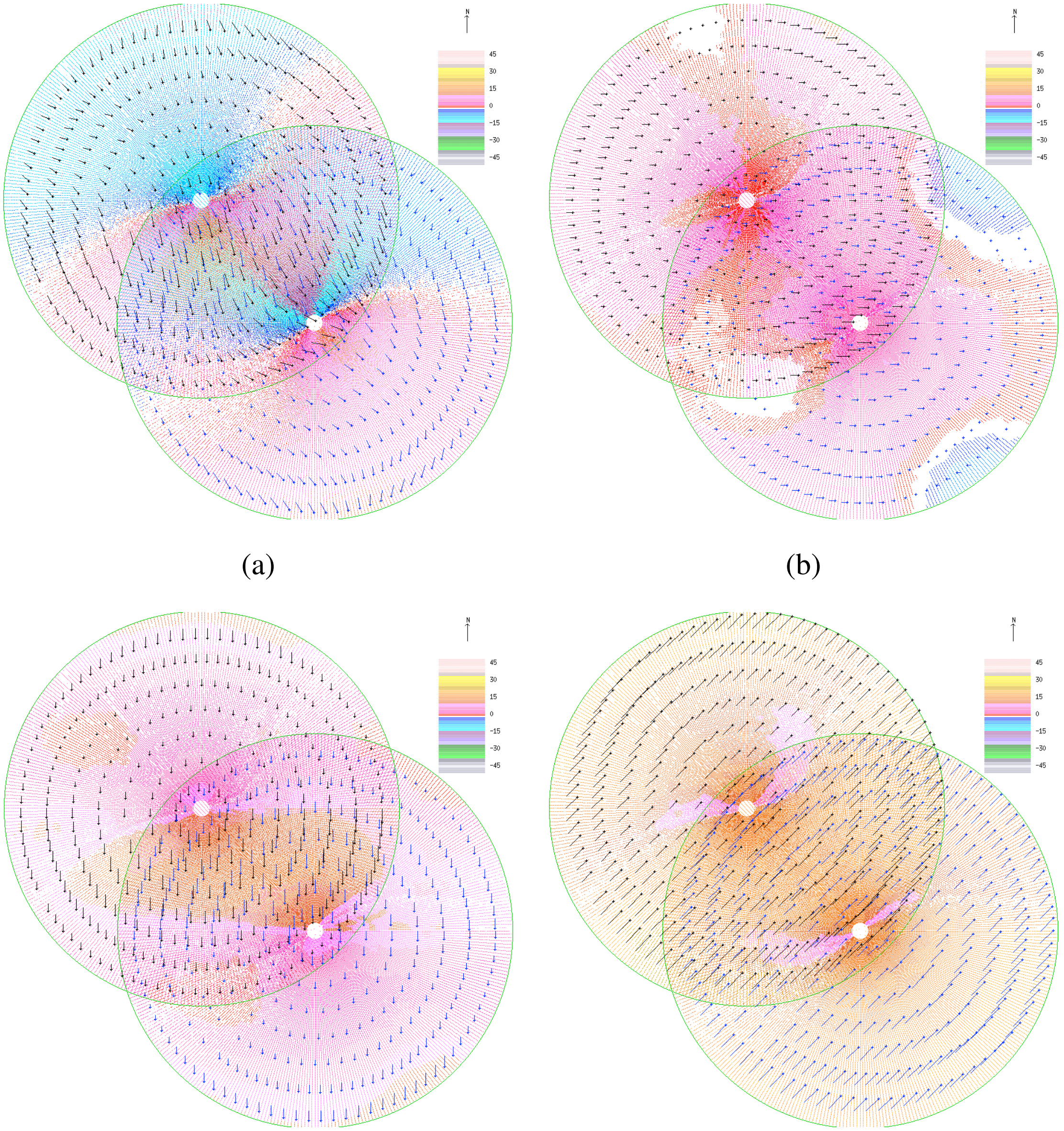
Figure B.38: The dual regularization retrieved synthetic velocity of group 2 at variation level $K = 0$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

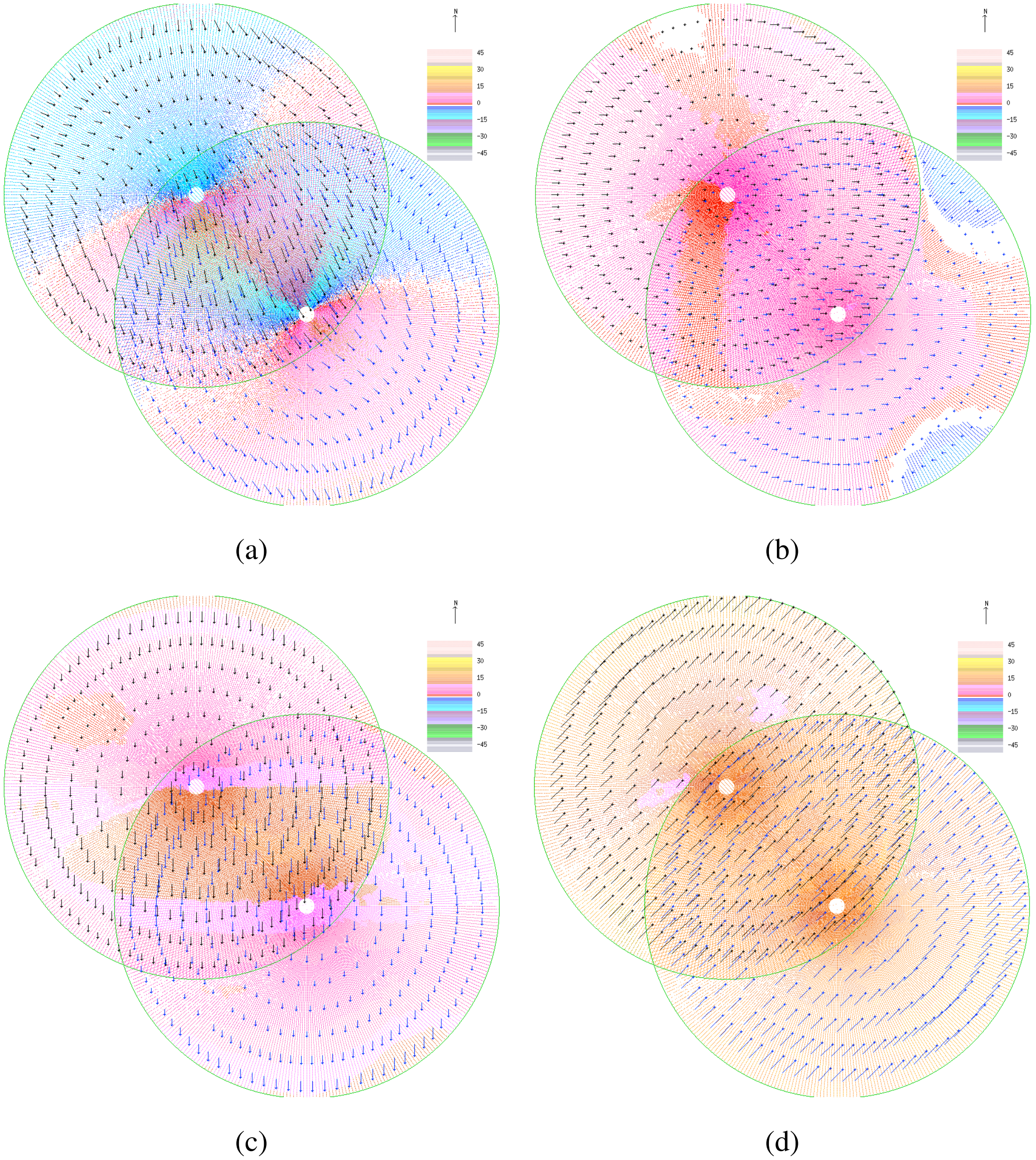
Figure B.39: The correct synthetic velocity of group 2 at variation level $K = 5$: (a) the correct UV flow, (b) the correct U component, (c) the correct V component and (d) the correct W component.



(a) (b)

(c) (d)

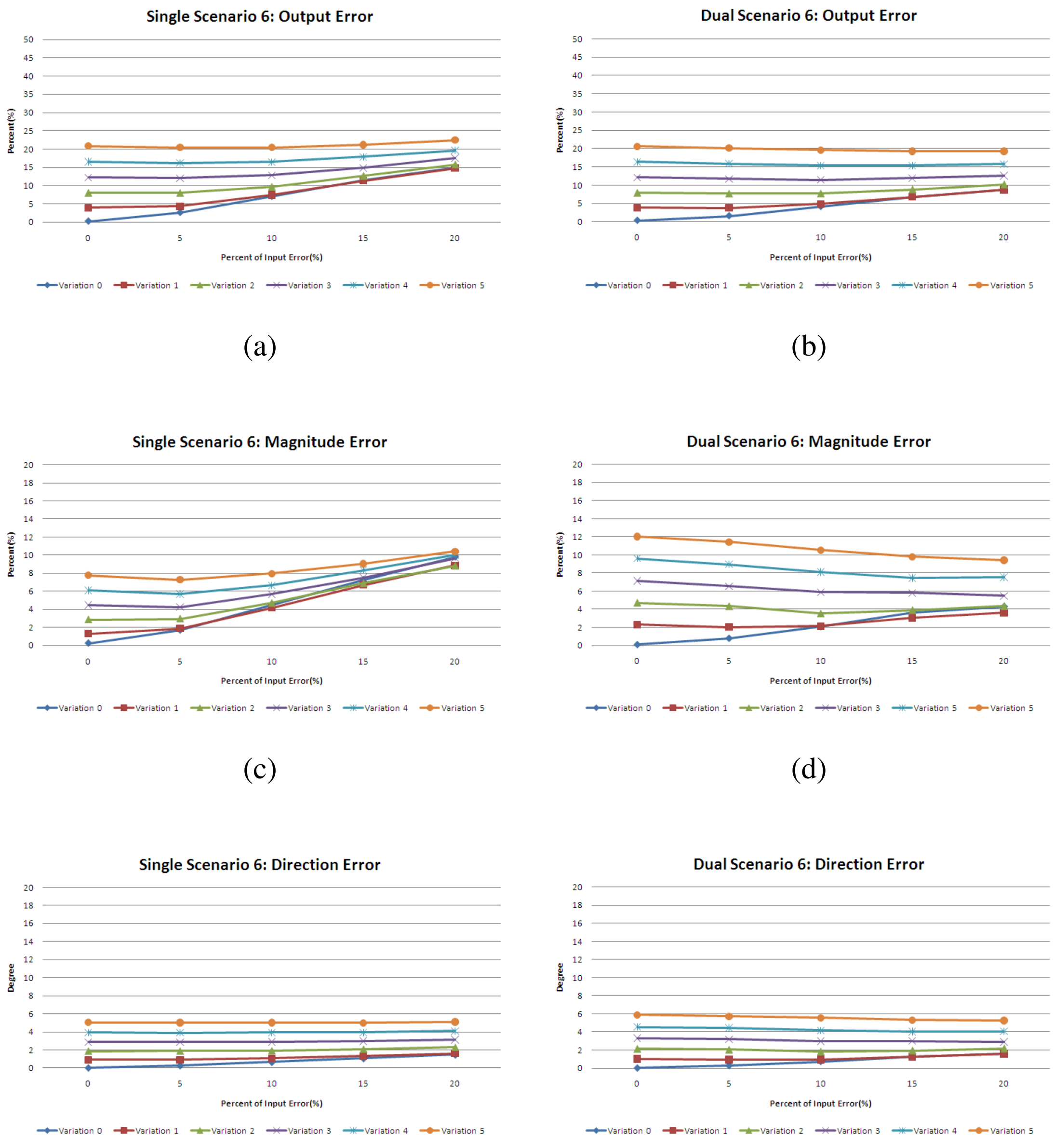
Figure B.40: The single regularization retrieved synthetic velocity of group 2 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.



(a) (b)

(c) (d)

Figure B.41: The dual regularization retrieved synthetic velocity of group 2 at variation level $K = 5$ and noise level $L = 4$: (a) the retrieved UV flow, (b) the retrieved U component, (c) the retrieved V component and (d) the retrieved W component.

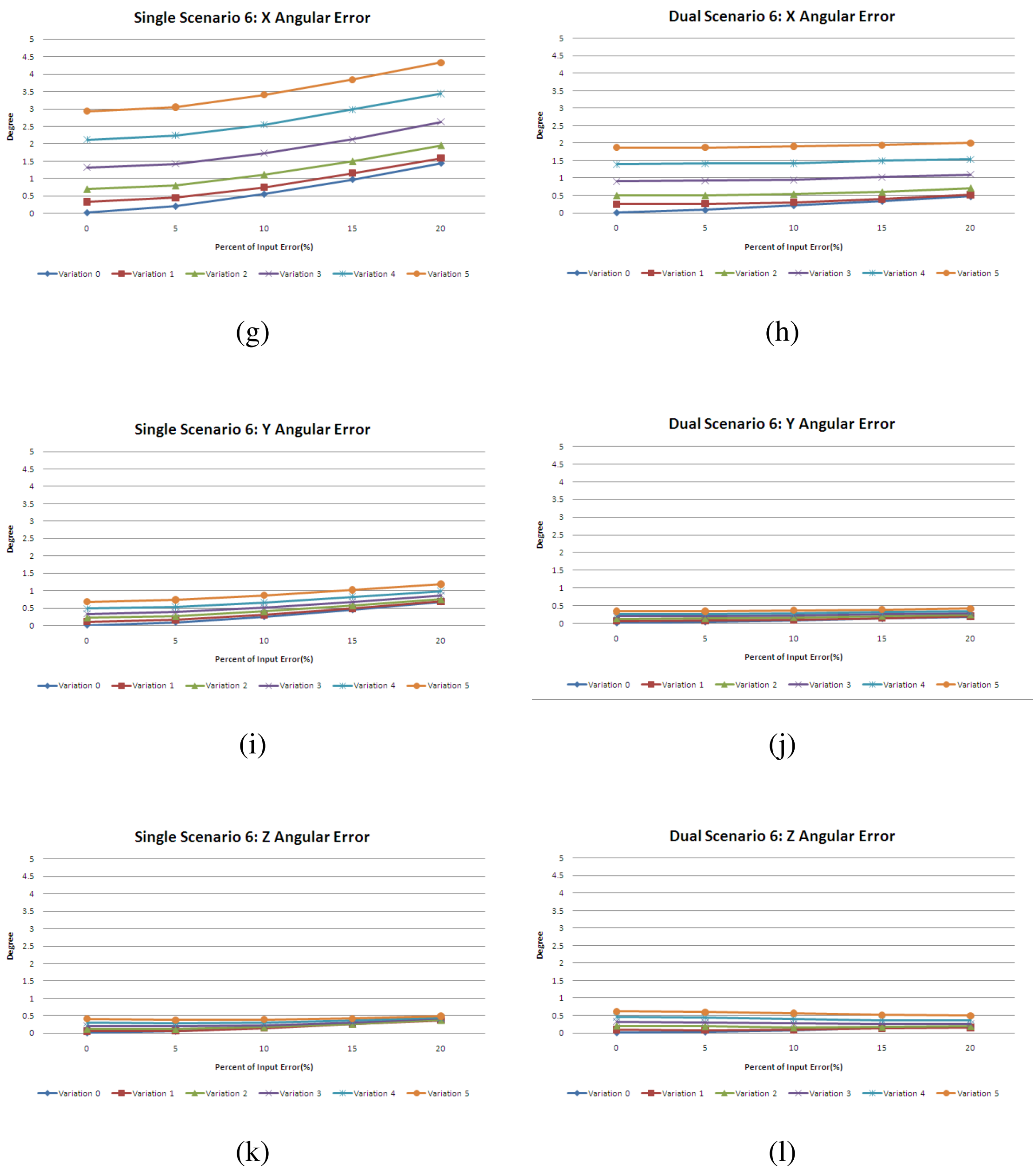


(a) (b)

(c) (d)

(e) (f)

Figure B.42: Result Analysis of Synthetic Data Group 6 using regularization method: (5.0, 10.0, 20.0). The Single Retrieval (a), (c), (e) and Dual Retrieval (b), (d), (f).



(g) (h)

(i) (j)

(k) (l)

Figure B.42: Result Analysis of Synthetic Data Group 6 using regularization method: (5.0, 10.0, 20.0). The Single Retrieval (g), (i), (k) and Dual retrieval (h), (j), (l).

Appendix C

Pseudo Code

Algorithm 1 FloodFill Algorithm using Stack

```
Require: Give one radar data point x
Ensure: Find all the radar data points from all the radars connected to it.
  if x.status = NOT_CHECK then
    if x.reflectivity $\geq$ Reflectivity_Threshold then
      x.stormNumber $\leftarrow$ CurrentStormNumber
      push(x)
    else
      x.stormNumber $\leftarrow$ NOT_STORM
    end if
    while stack is not empty do
      y $\leftarrow$ pop()
      for each radar i that has data available do
        for each data point z in the neighborhood of y from radar i do
          if z.status = NOT_CHECK then
            if z.reflectivity $\geq$ Reflectivity_Threshold then
              z.stormNumber $\leftarrow$ CurrentStormNumber
              push(z)
            else
              z.stormNumber $\leftarrow$ NOT_STORM
            end if
          end if
        end for
      end for
    end while
  end if
```

Algorithm 2 Pseudo Storm Detection Algorithm

```
Require: Have three continuous images with storms detected.
Ensure: Obtain the two pseudo storm numbers for each storm
  for each storm $i$ in the current image do
    for each storm $j$ in the previous image do
      if i.center is in j then
        PreviousStack[j].push(i)
      end if
    end for
    for each storm $k$ in the next image do
      if i.center is in k then
        NextStack[k].push(i)
      end if
    end for
  end for
  for each storm $j$ in the previous image do
    if PreviousStack[j].length $\geq 2$ then
      PreviousPseudoNumber $\leftarrow$ PreviousPseudoNumber+1
      while PreviousStack[j].length $\geq 0$ do
        i $\leftarrow$ PreviousStack[j].pop()
        i.PreviousStorm $\leftarrow$ j
      end while
    end if
  end for
```

```
for each storm $k$ in the next image do
  if NextStack[k].length ≥ 2 then
    NextPseudoNumber ← NextPseudoNumber+1
    while NextStack[k].length ≥ 0 do
      i ← NextStack[k].pop()
      i.NextStorm ← k
    end while
  end if
end for
```

Algorithm 3 Pseudo Storm Tracking Selection Algorithm SelectTracks(AdjcencySet)

```
Require: Have all the available adjacencies saved in AdjcencySet
Ensure: Select adjacencies to generate suitable tracks
  for each AdjcencyList in AdjacencySet do
    adjacencyLeft ← AdjcencyList.num
    while adjacencyLeft > 0 do
      Find the available adjacency that has highest compatibility value, adj
      call function SelectAdjacency (adj)
      Collect the number of adjacencie left in AdjacencyList, m
      adjacencyLeft ← m
    end while
  end for
```

Algorithm 4 Adjacency Selection Algorithm SelectAdj(*adj*)

Require: change *adj*'s status and process all its disparities

Ensure: *adj* and all its diaparities must be available

```
Adj.status ← CHOSEN
for each disparity, disp, in adj's preceding disparity list do
   disp.status ← CHOSEN
end for
for each disparity, disp, in adj's following disparity list do
   disp.status ← CHOSEN
end for
for each disparity, disp, in adj's preceding disparity list do
   Call function SelectDisp(disp, adj)
end for
for each disparity, disp, in adj's following disparity list do
   Call function SelectDisp(disp, adj)
end for
```

Algorithm 5 Disparity Selection Algorithm SelectDisp(*disp*, *adj*)

```
Require: disp contains a list of adjacencies that use it as preceding or following disparity
Ensure: disp is adopted by adj
  if disp is adopted by adj as preceding disparity then
    for each adjacency, a, that is not adj and uses disp as preceding do
      if a.status = CHOSEN then
        Report error
      end if
    end for
    if no adjacency that uses disp as following disparity has been chosen then
      Find the adjacency with highest compatibility value that uses disp as following disparity, a
      Call function SelectAdj(a)
    end if
  end if
  if disp is adopted by adj as following disparity then
    for each adjacency, a, that is not adj and uses disp as following do
      if a.status = CHOSEN then
        Report error
      end if
    end for
    if no adjacency that uses disp as preceding disparity has been chosen then
      Find the adjacency with highest compatibility value that uses disp as preceding disparity, a
      Call function SelectAdj(a)
    end if
  end if
```

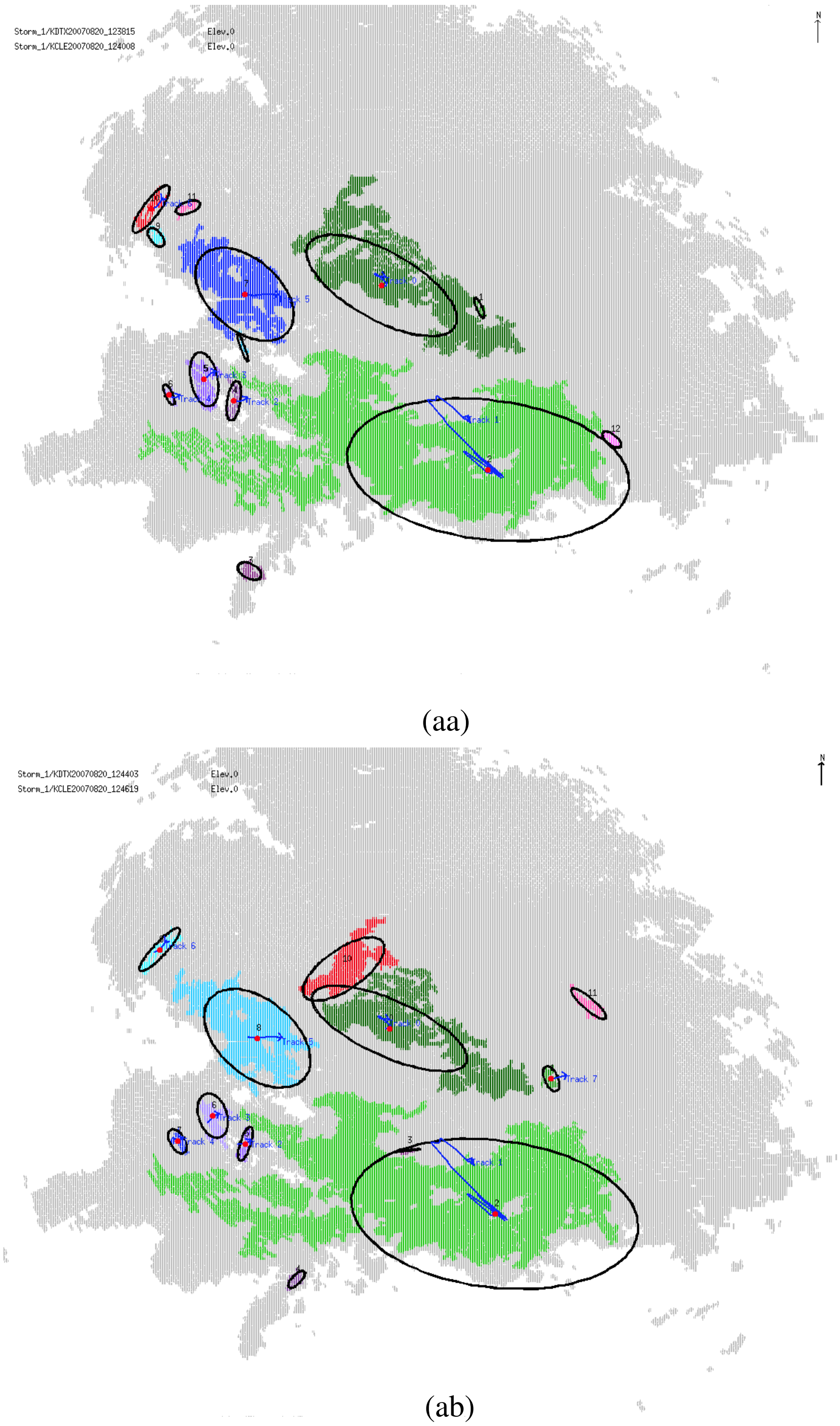
Appendix D

Tracking Results with Multiple Doppler Radars on August 20^{th}, 2007

D.1 Image Sequence 1

The first group of figures displays the tracking results from the Detroit and Cleveland Doppler data on August 20^{th}, 2007 using the original relaxation labeling algorithm.

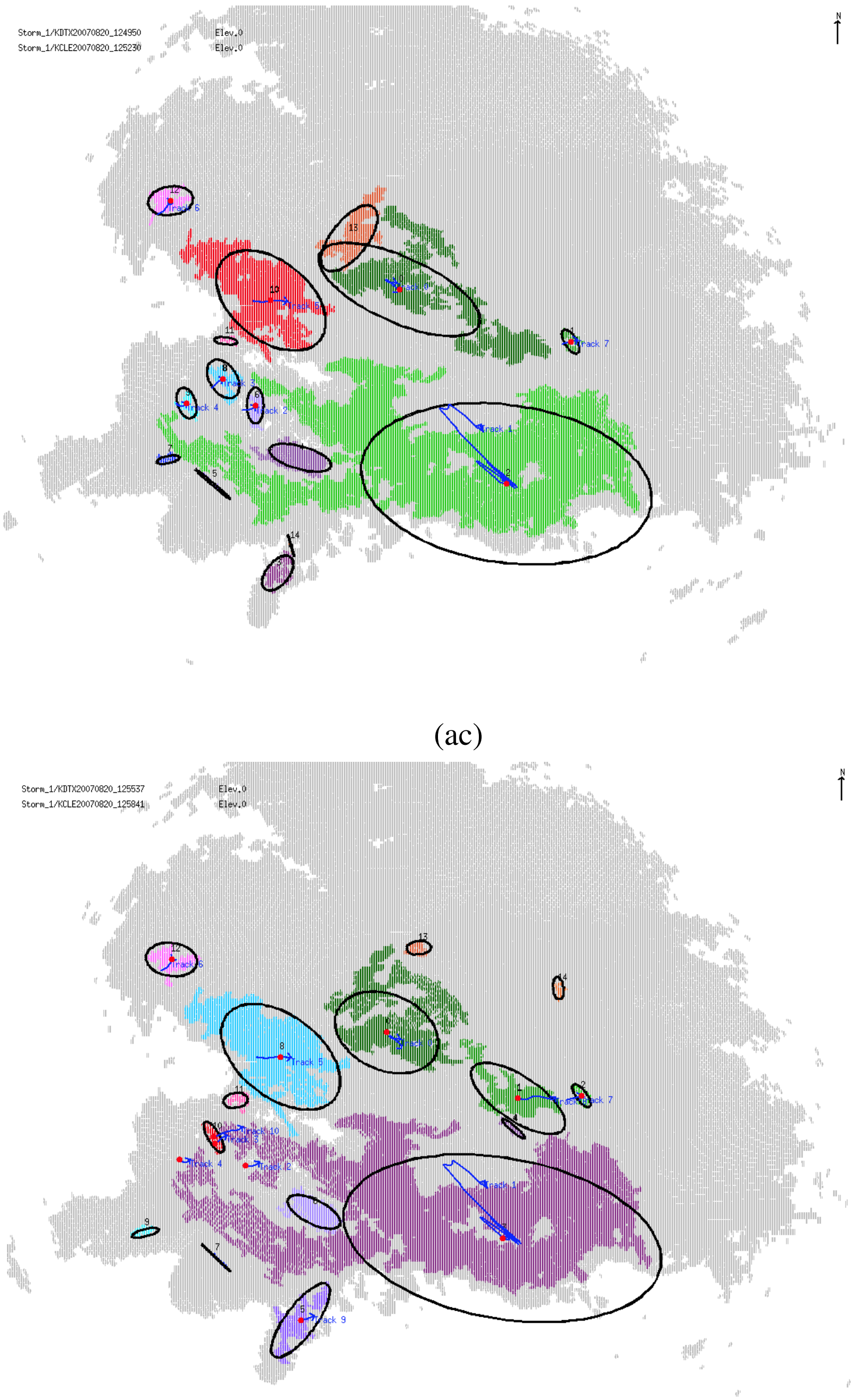
In each image, the large grey-coloured area indicates non-zero dBZ but no storm. All of the other colours show different recognized storms. The ellipses drawn on the storms represent the size, location of the center and orientation of the ellipsoids representing the storms. The blue curves show the complete tracks of the storms and the red dots on the tracks show the current positions of the storms on the tracks.



(aa)

(ab)

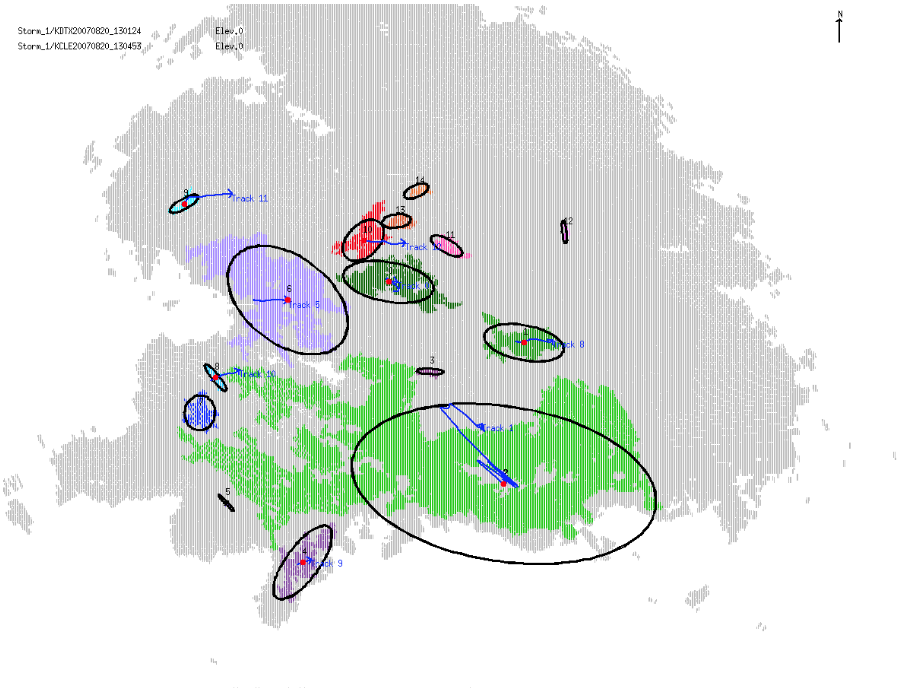
Figure D.1: The tracks on Images (*aa*) and (*ab*) of the 27 images using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007



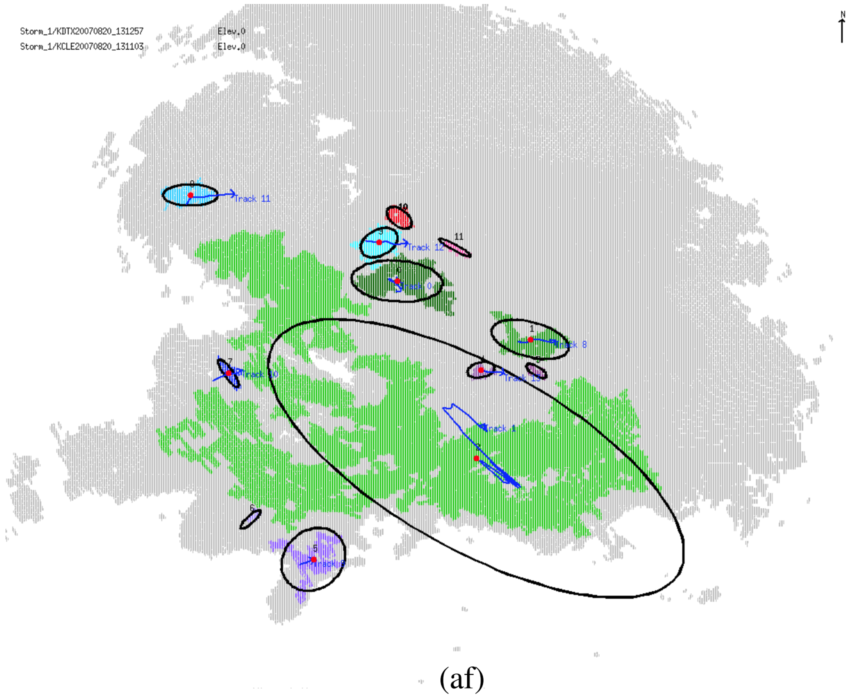
(ac)

(ad)

Figure D.1: The tracks on Images (*ac*) and (*ad*) of the 27 images using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20^{th}, 2007

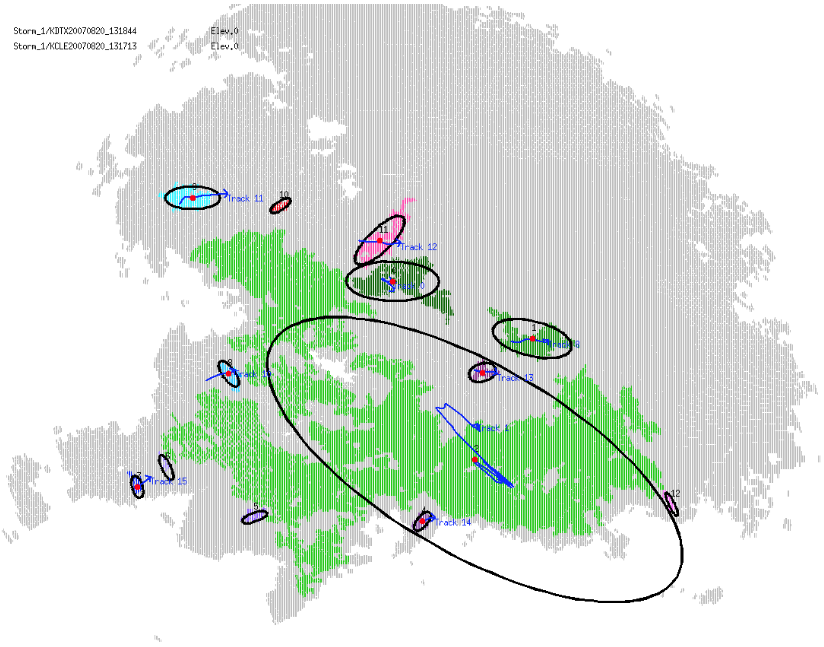


(ae)

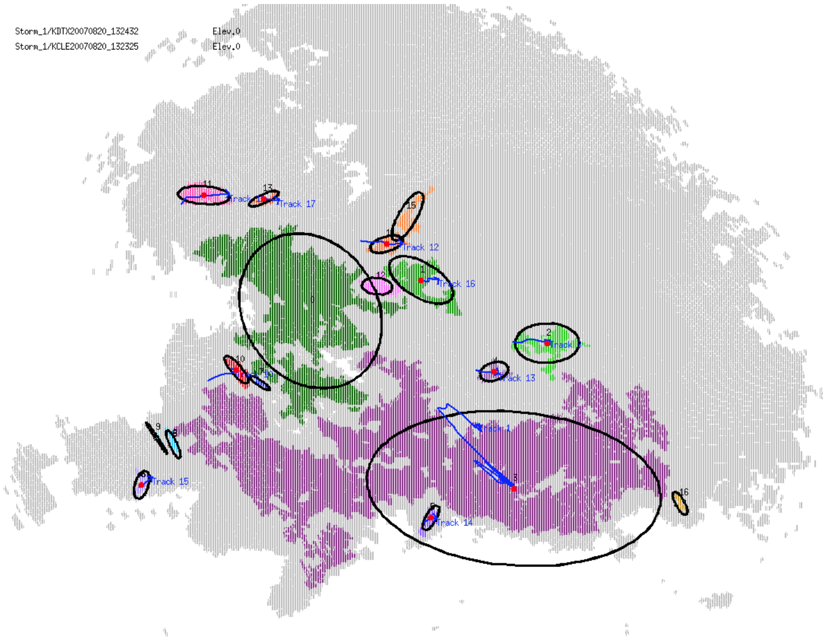


(af)

Figure D.1: The tracks on Images (ae) and (af) of the 27 images using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007

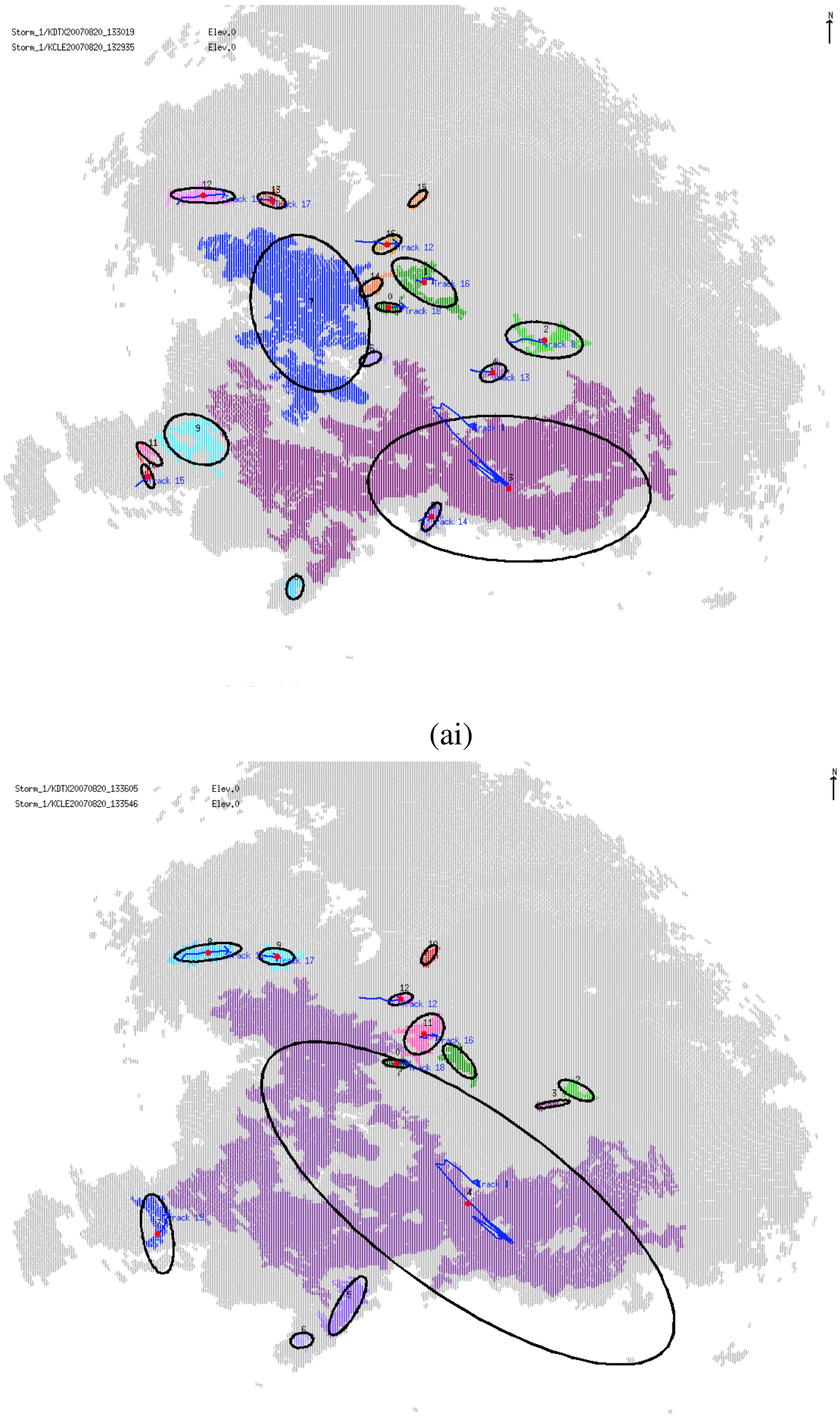


(ag)



(ah)

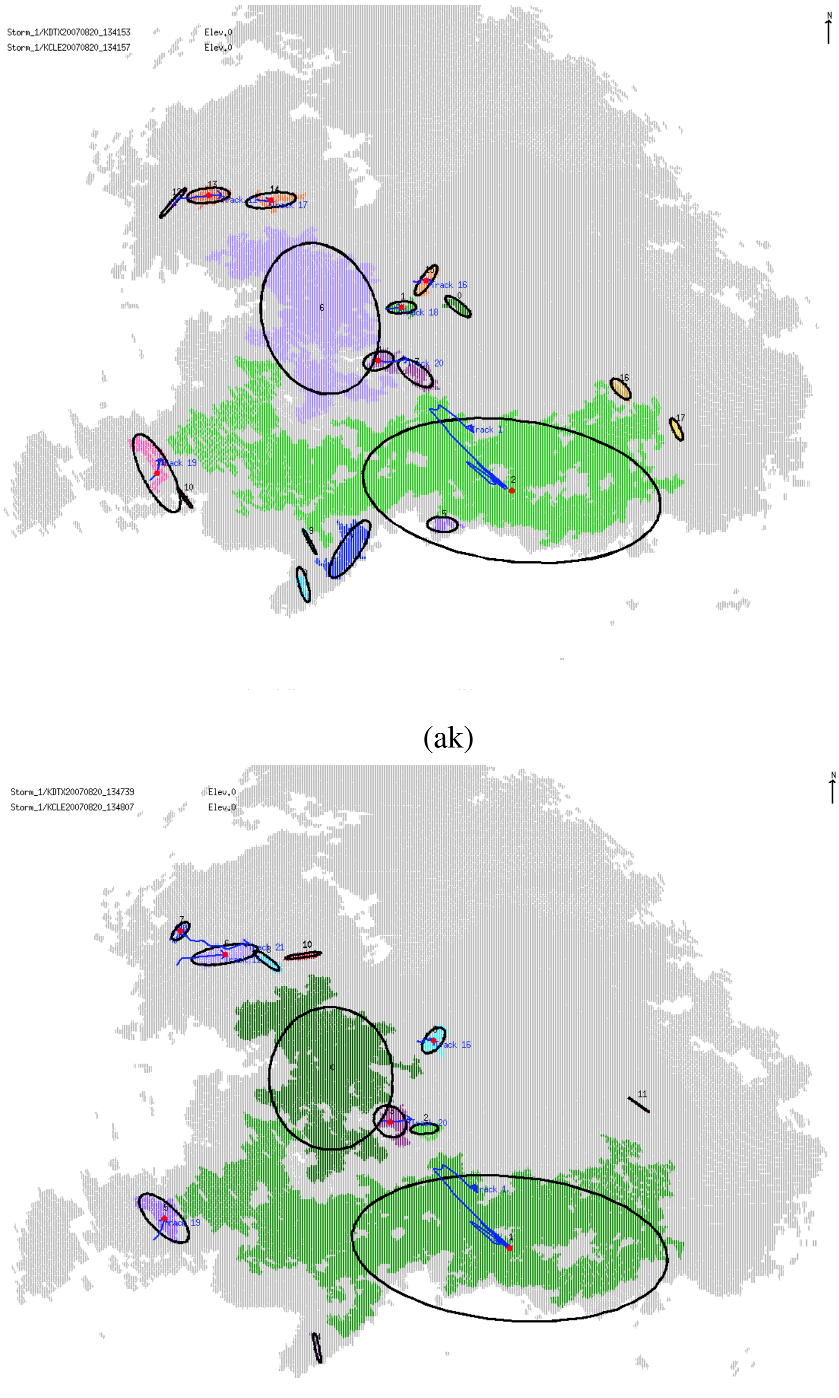
Figure D.1: The tracks on Images (*ag*) and (*ah*) of the 27 images using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007



(ai)

(aj)

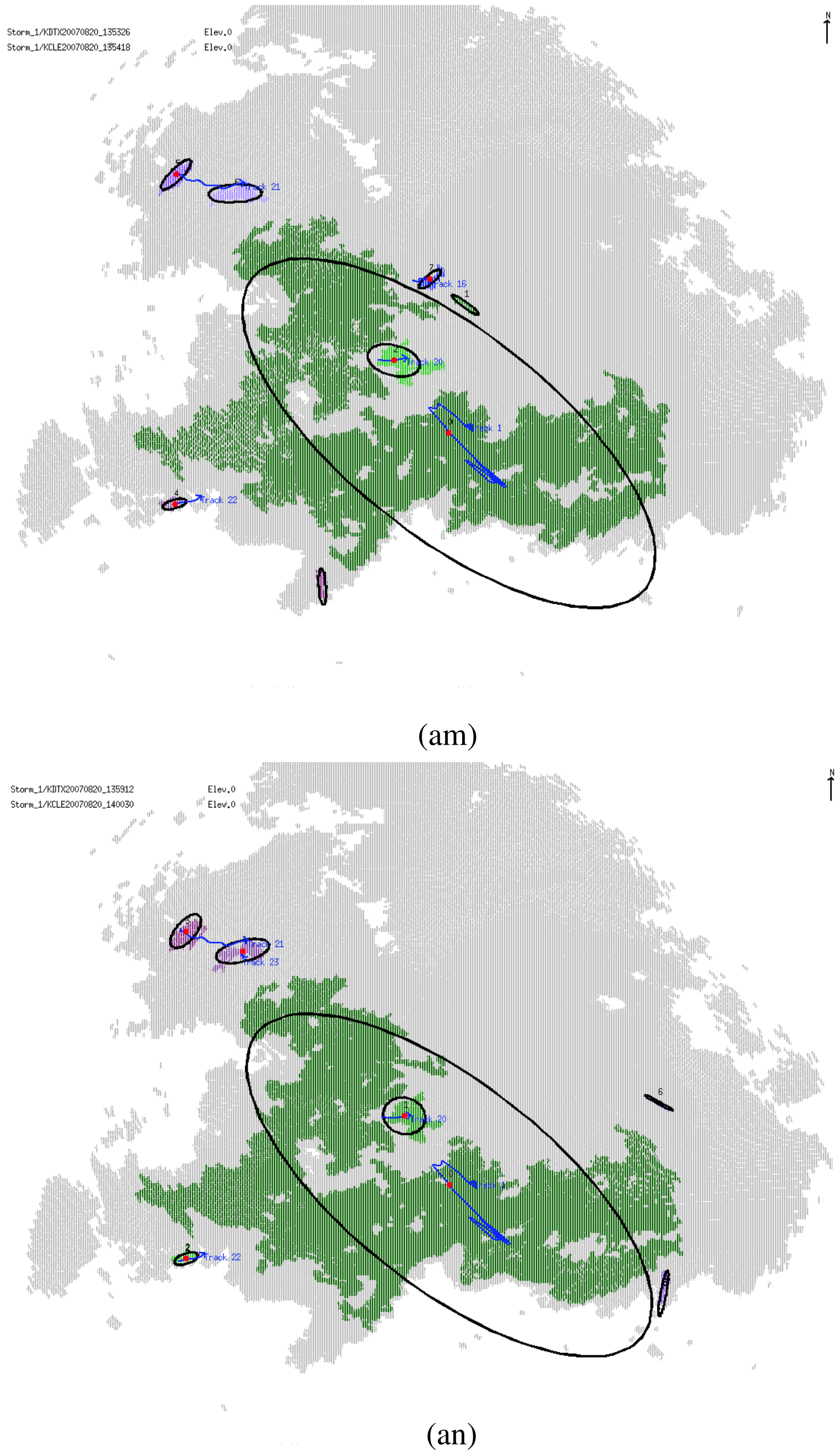
Figure D.1: The tracks on Images (ai) and (aj) of the 27 images using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007



(ak)

(al)

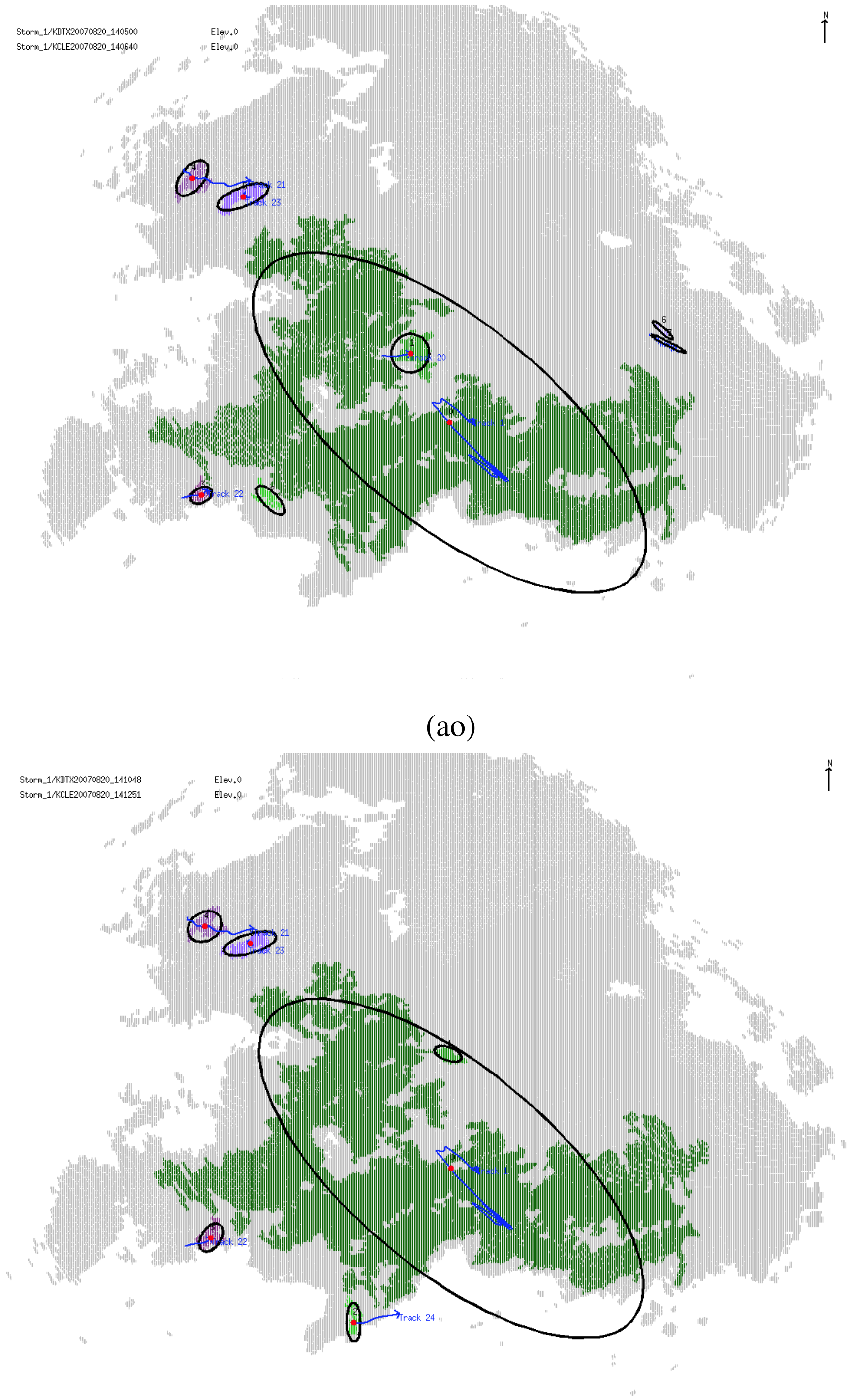
Figure D.1: The tracks on Images (*ak*) and (*al*) of the 27 images using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007



(am)

(an)

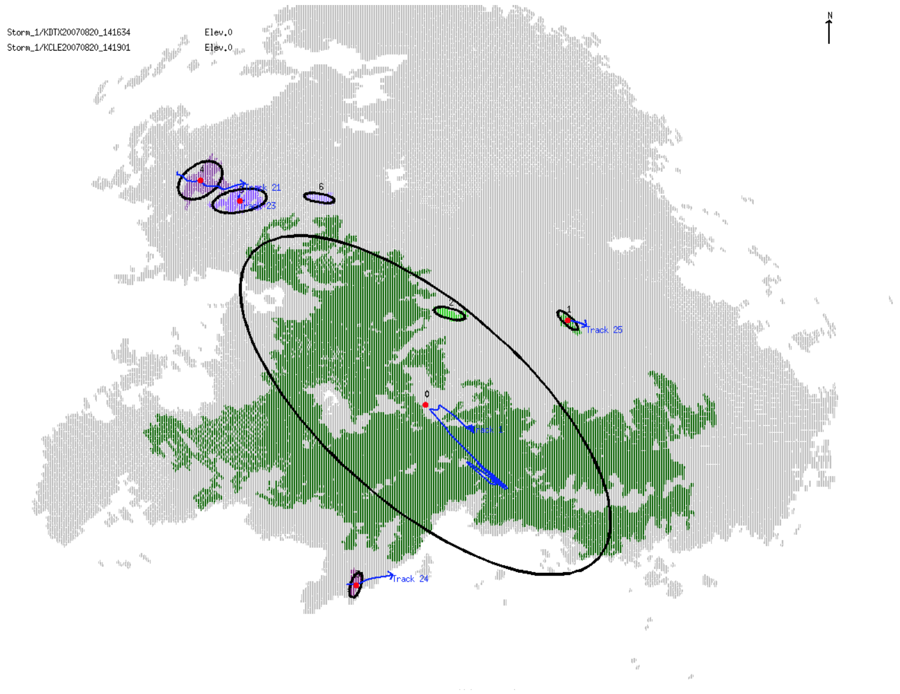
Figure D.1: The tracks on Images (*am*) and (*an*) of the 27 images using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20^{th}, 2007



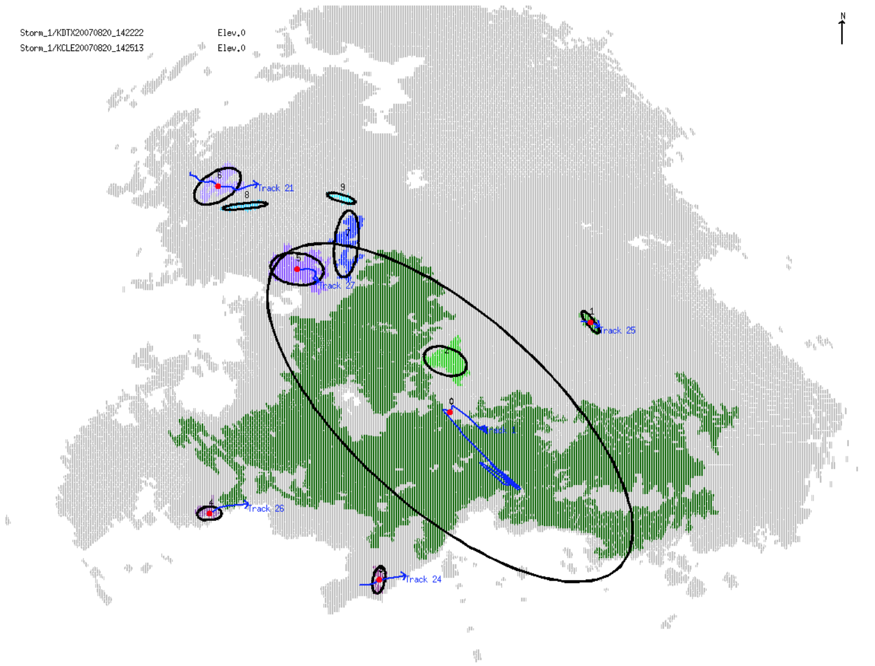
(ao)

(ap)

Figure D.1: The tracks on Images (*ao*) and (*ap*) of the 27 images using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007

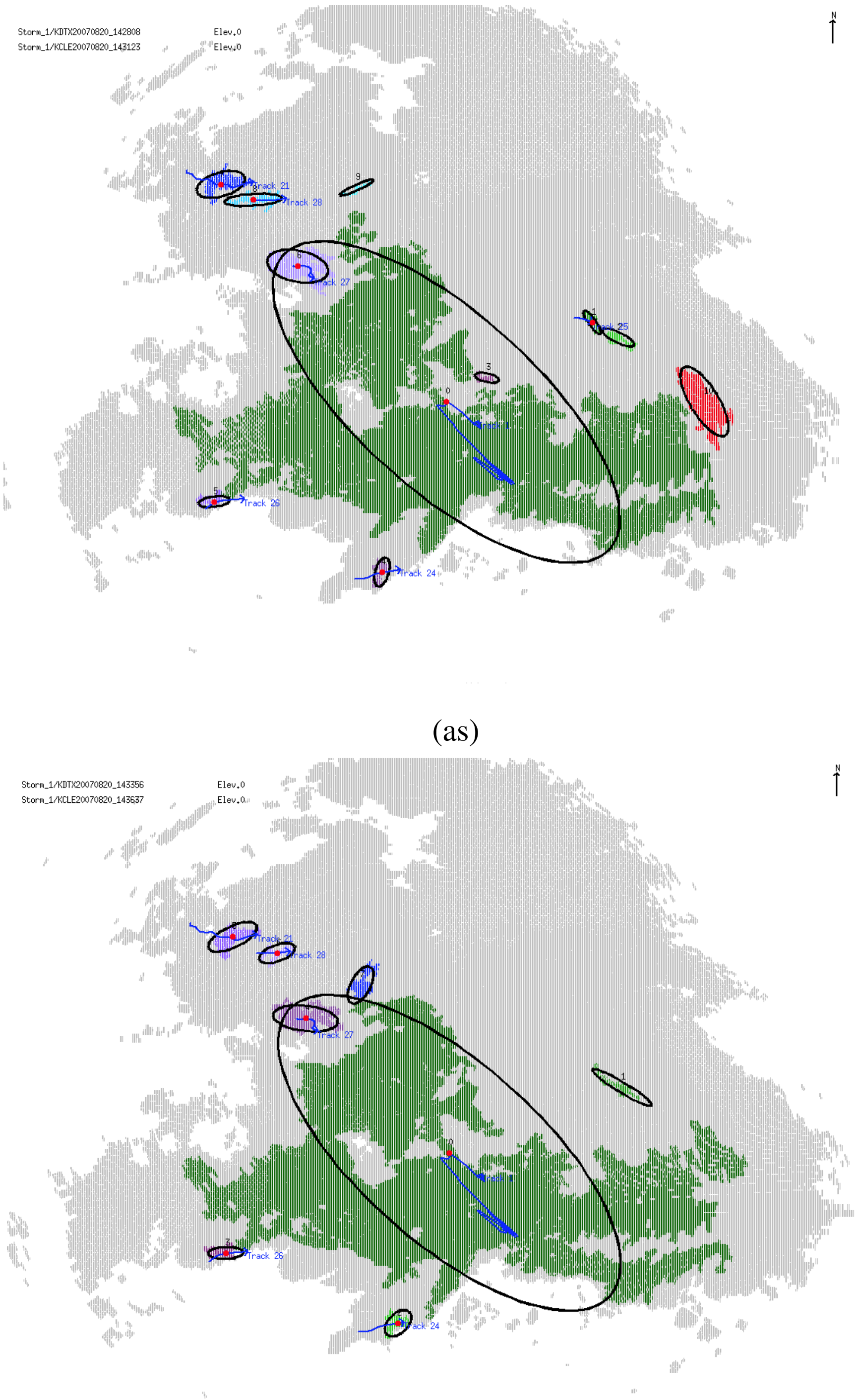


(aq)



(ar)

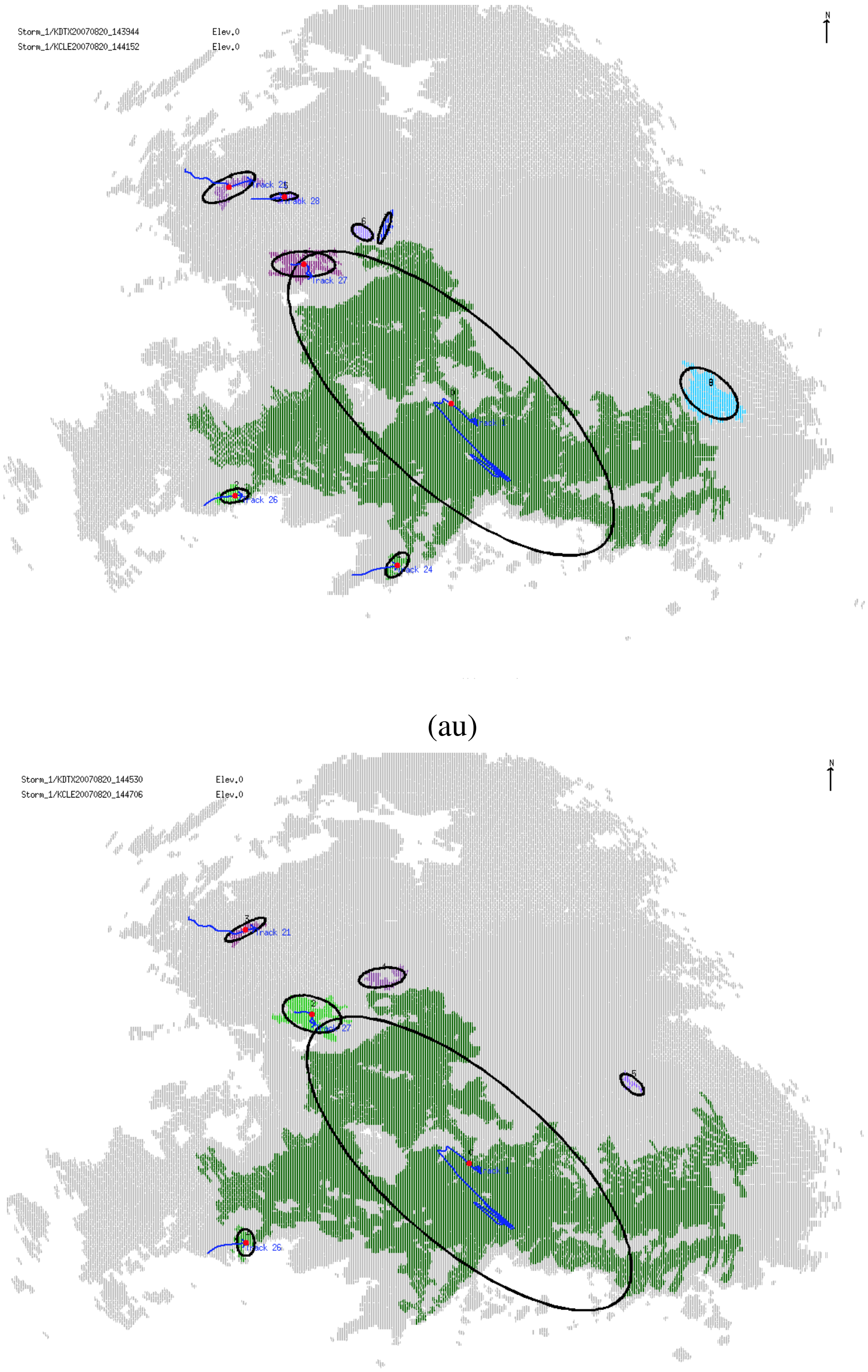
Figure D.1: The tracks on Images (*aq*) and (*ar*) of the 27 images using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20^{th}, 2007



(as)

(at)

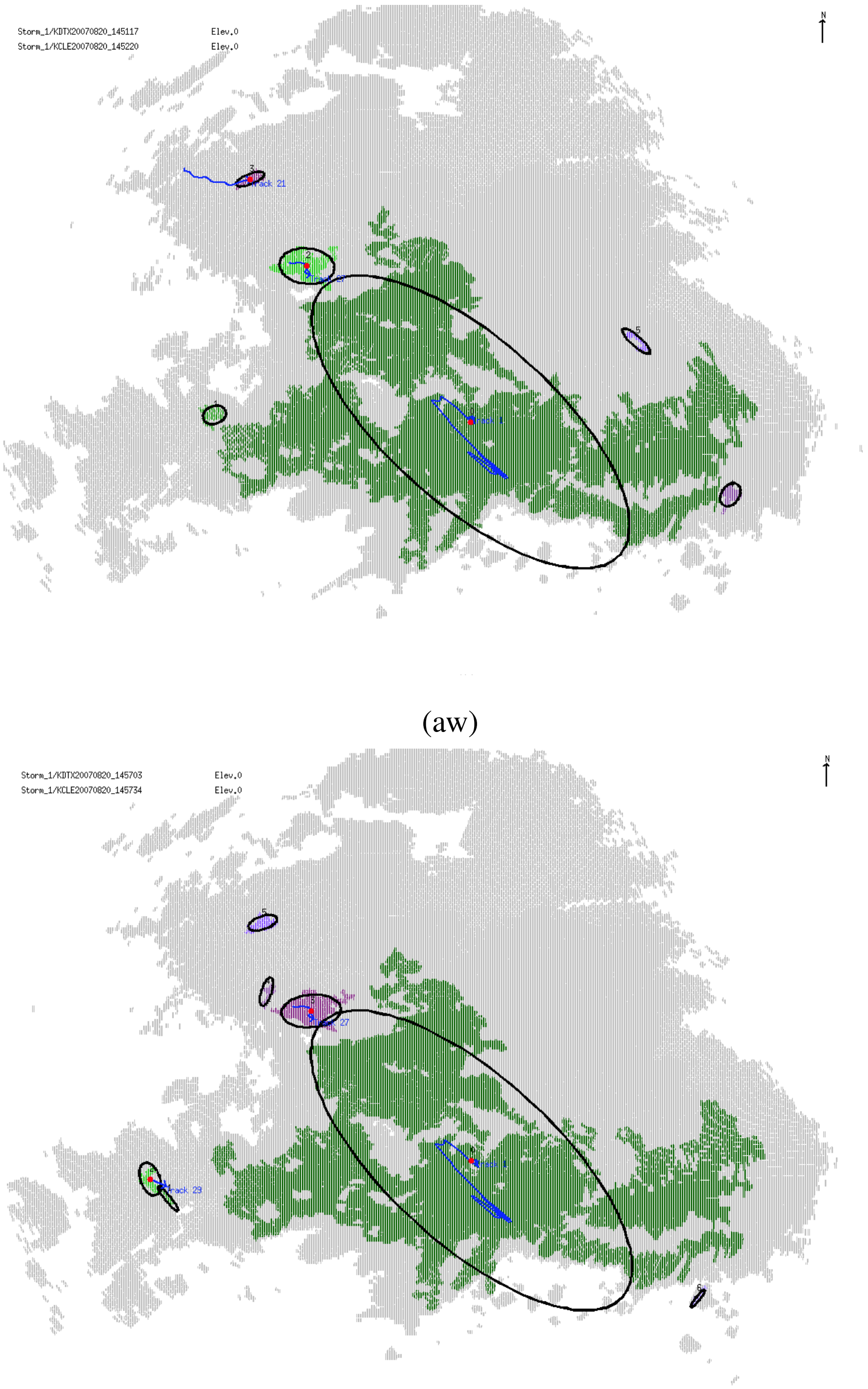
Figure D.1: The tracks on Images (*as*) and (*at*) of the 27 images using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20^{th}, 2007



(au)

(av)

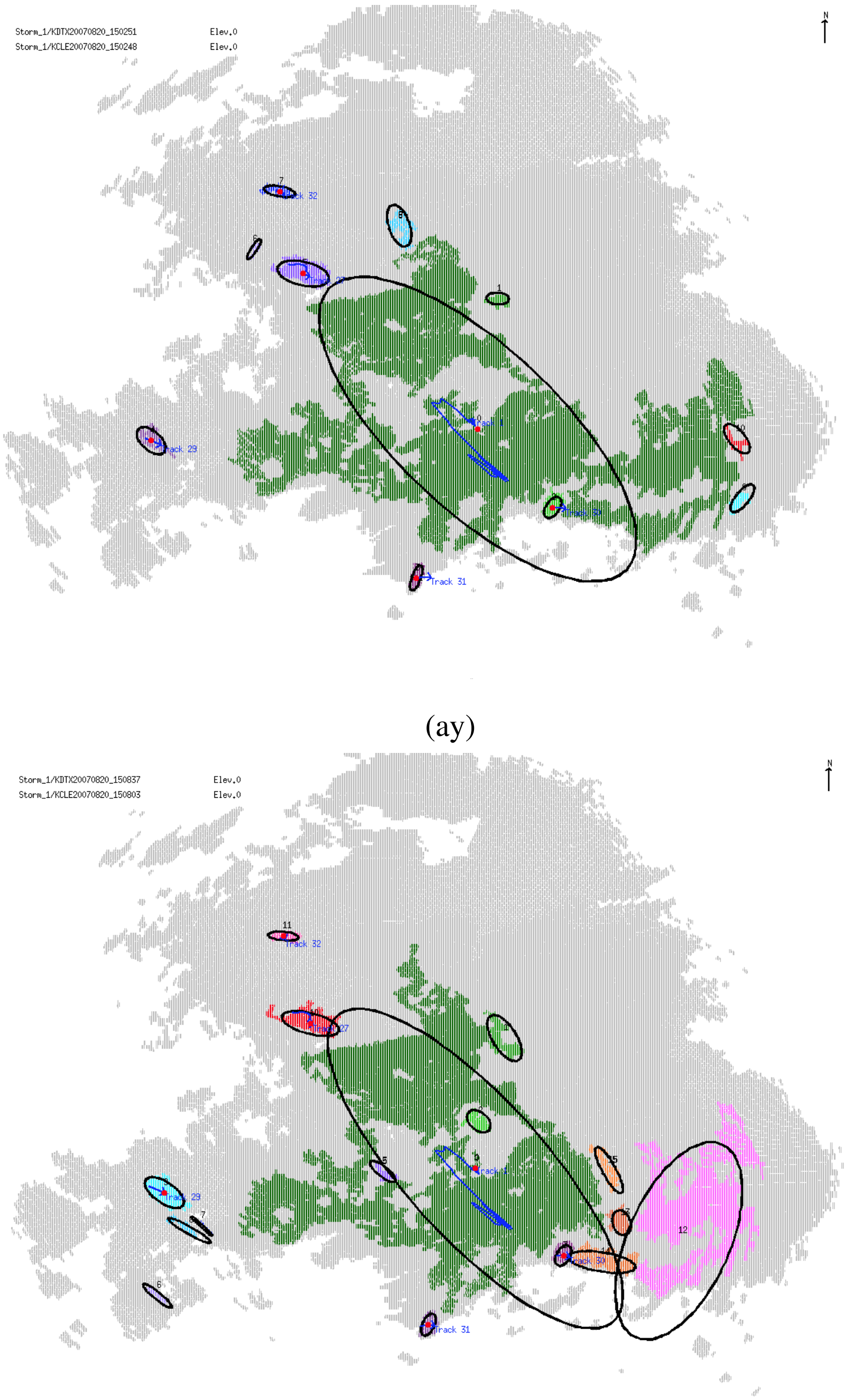
Figure D.1: The tracks on Images (*au*) and (*av*) of the 27 images using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007



(aw)

(ax)

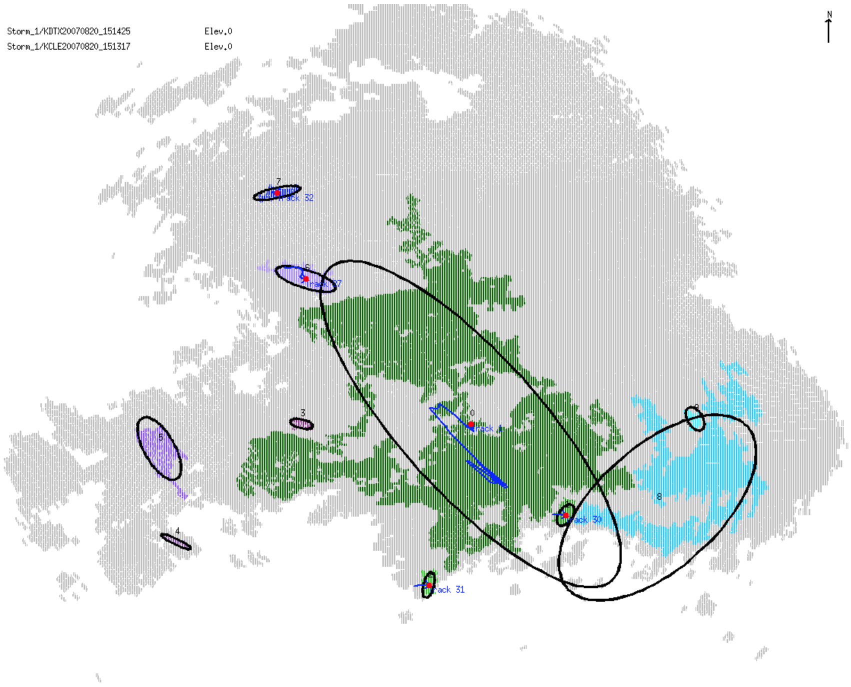
Figure D.1: The tracks on Images (*aw*) and (*ax*) of the 27 images using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20^{th}, 2007



(ay)

(az)

Figure D.1: The tracks on Images (*ay*) and (*az*) of the 27 images using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007

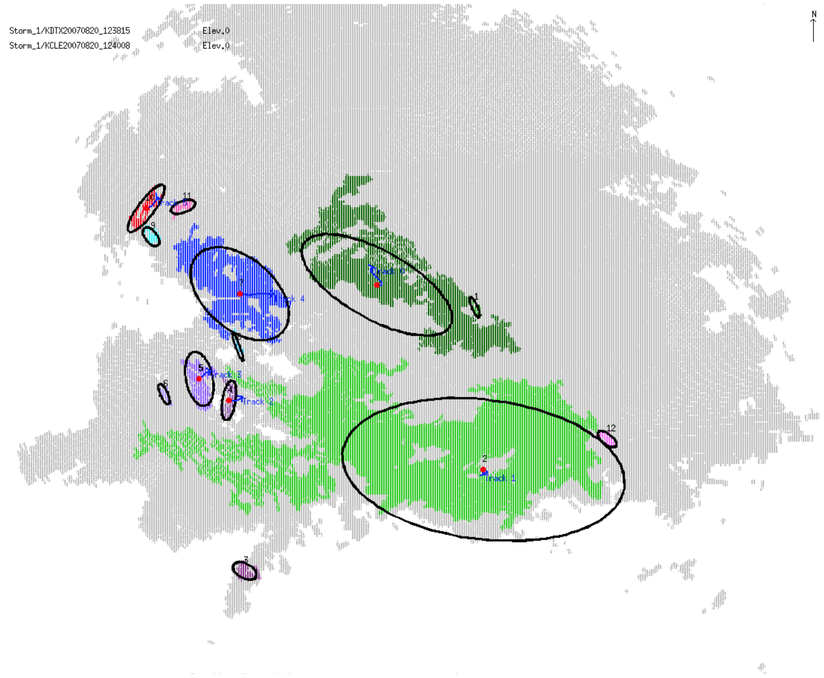


(ba)

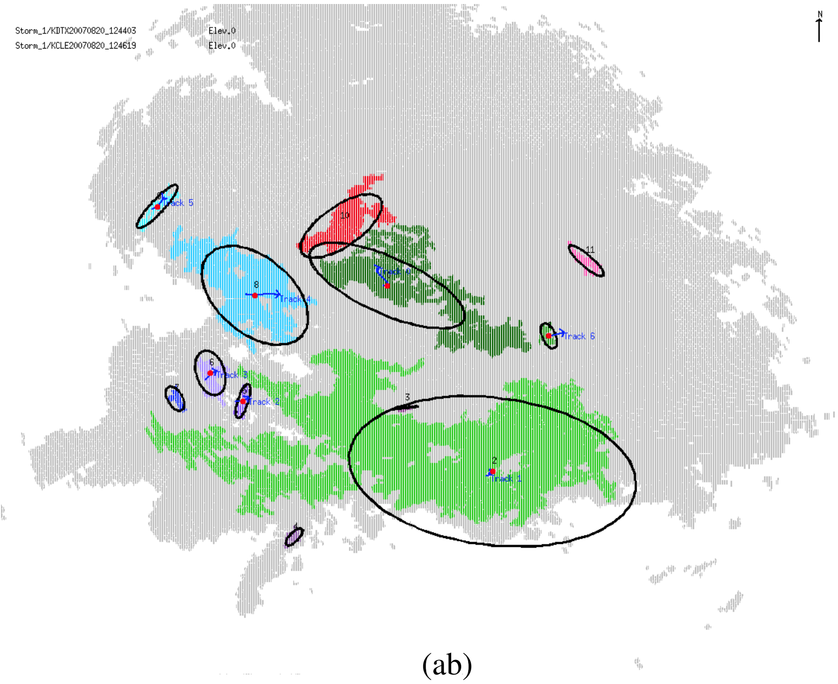
Figure D.1: The tracks on Image (*ba*) of the 27 images using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20^{th}, 2007

D.2 Image Sequence 2

The second group of figures displays the tracking results from the Detroit and Cleveland Doppler data on August 20^{th}, 2007 using the original relaxation labeling algorithm but implemented with pseudo storms.

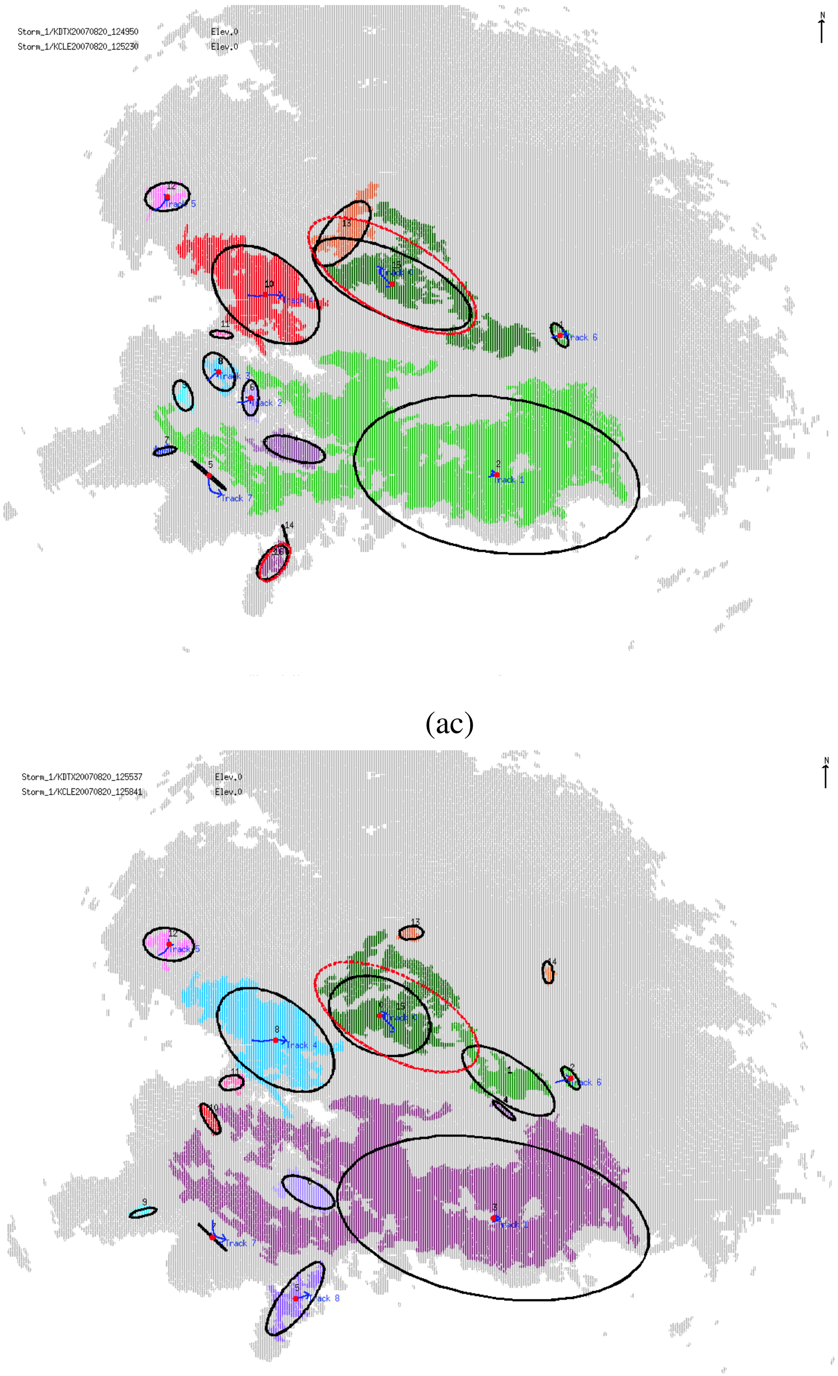


(aa)



(ab)

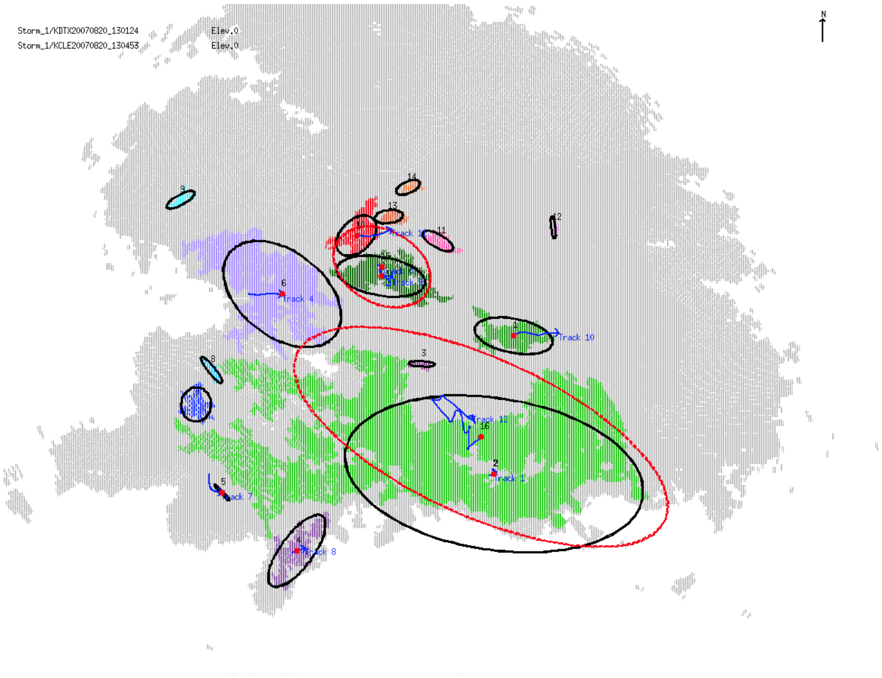
Figure D.2: The tracks on Images (*aa*) and (*ab*) of the 27 images with pseudo storms using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007



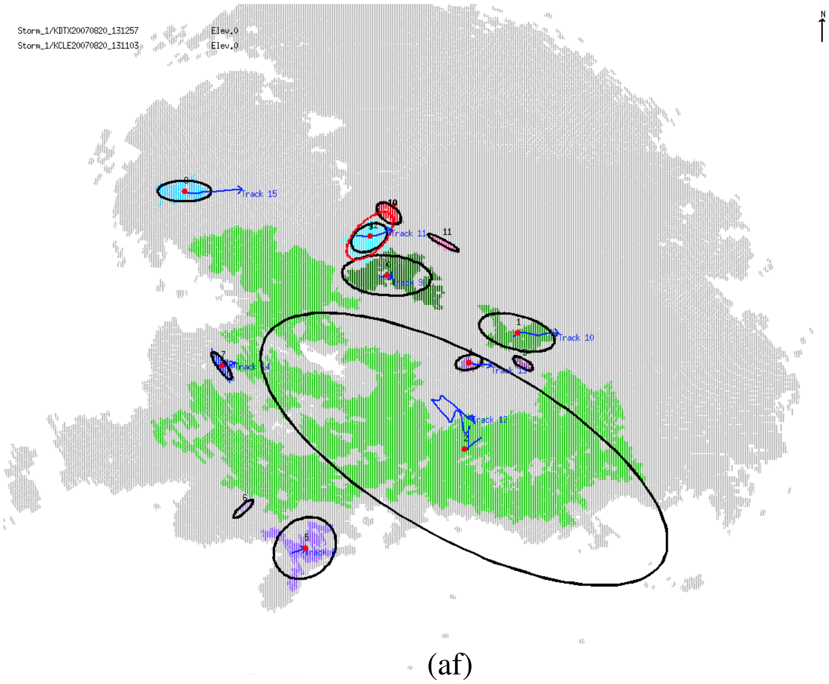
(ac)

(ad)

Figure D.2: The tracks on Images (*ac*) and (*ad*) of the 27 images with pseudo storms using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007

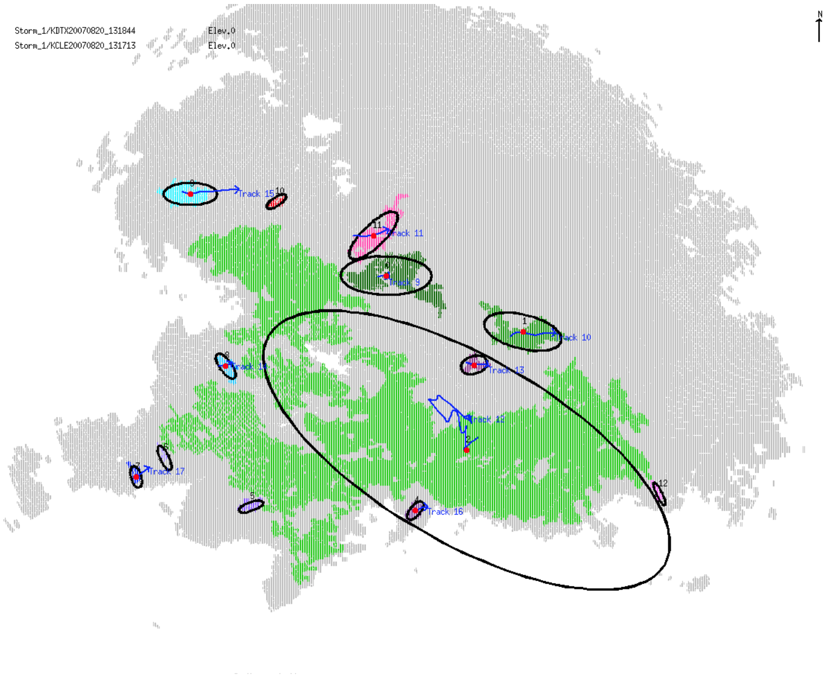


(ae)

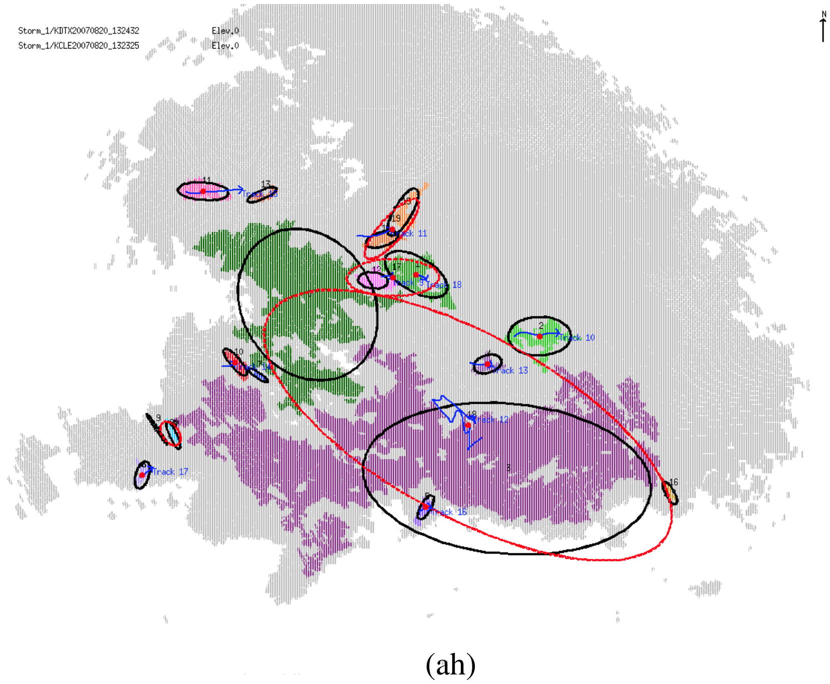


(af)

Figure D.2: The tracks on Images (ae) and (af) of the 27 images with pseudo storms using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007

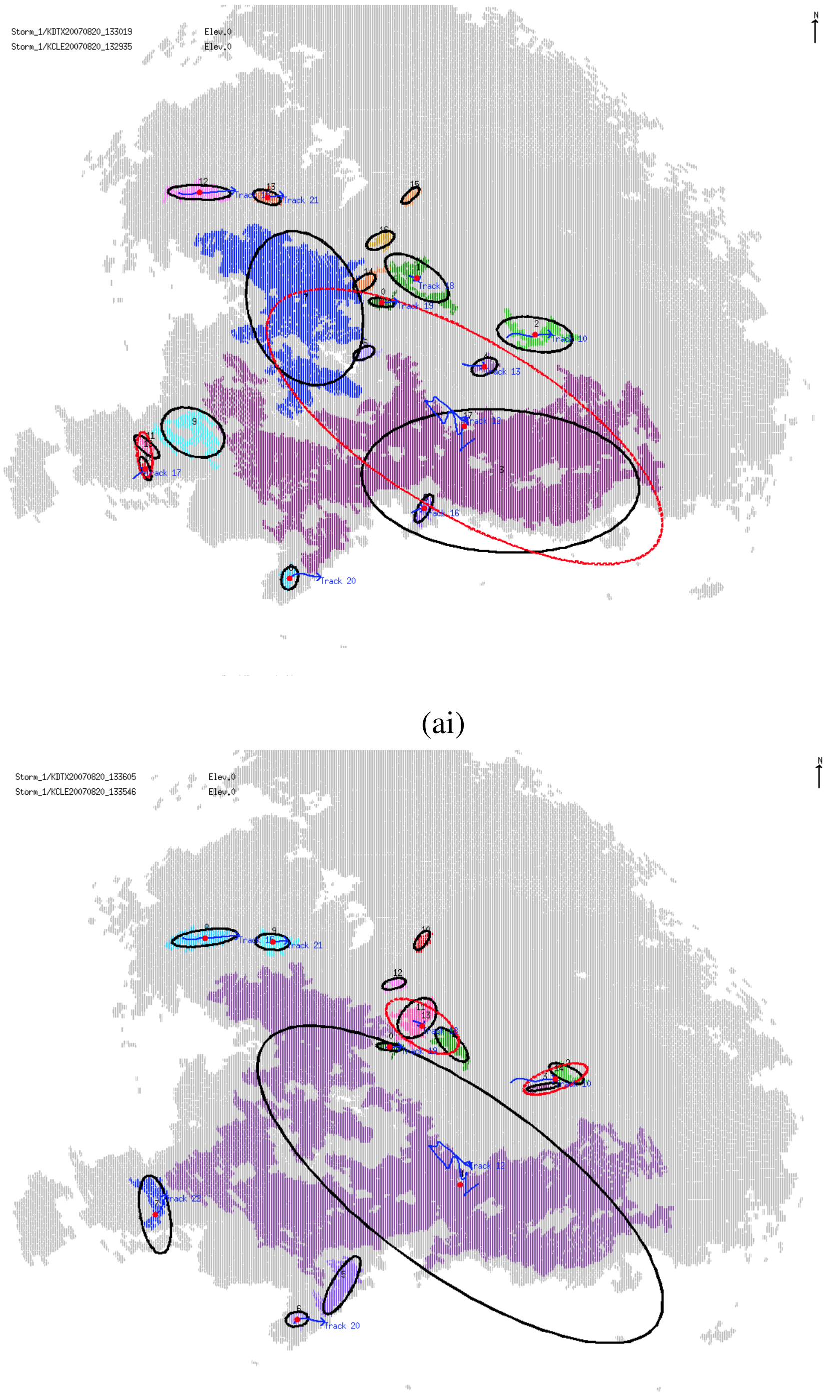


(ag)



(ah)

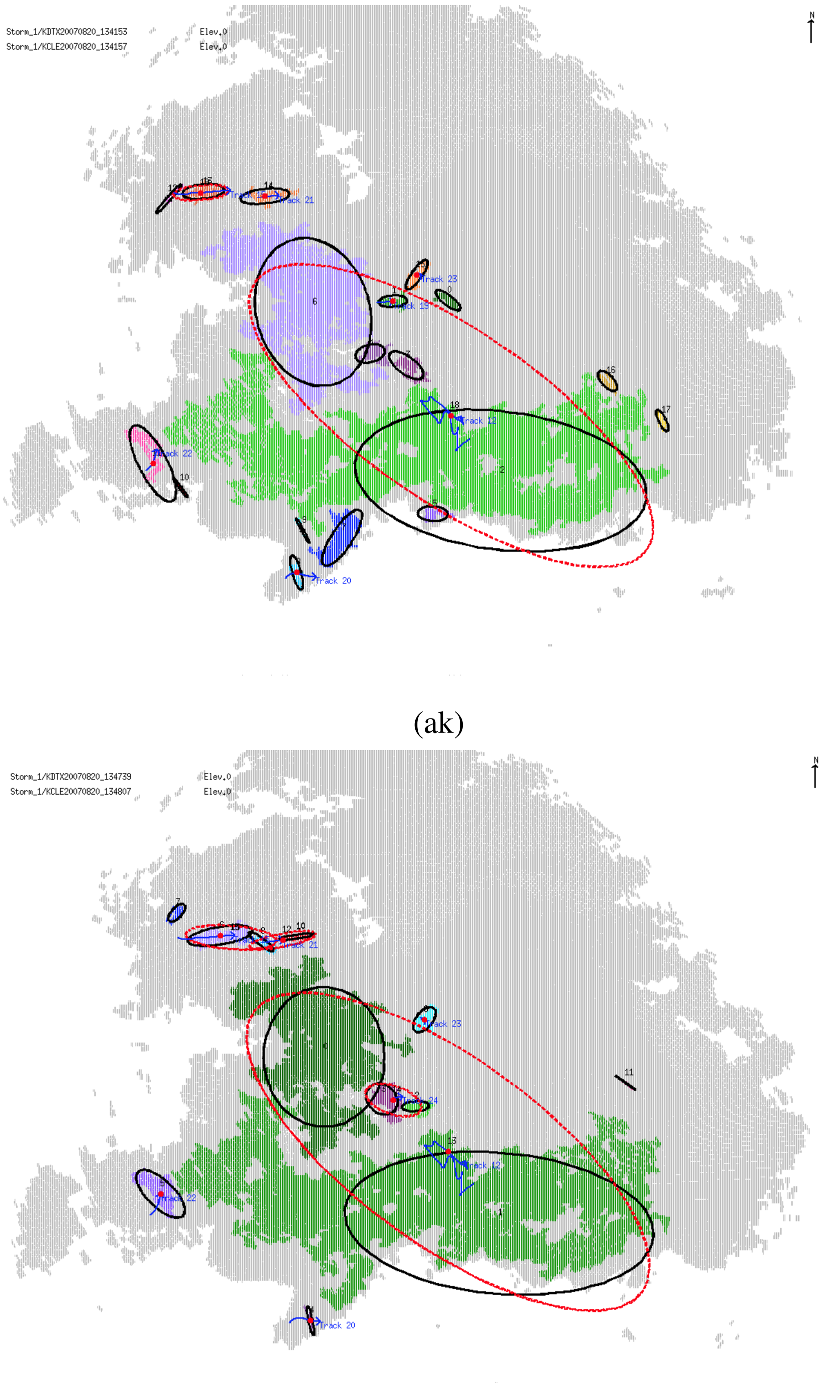
Figure D.2: The tracks on Images (*ag*) and (*ah*) of the 27 images with pseudo storms using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007



(ai)

(aj)

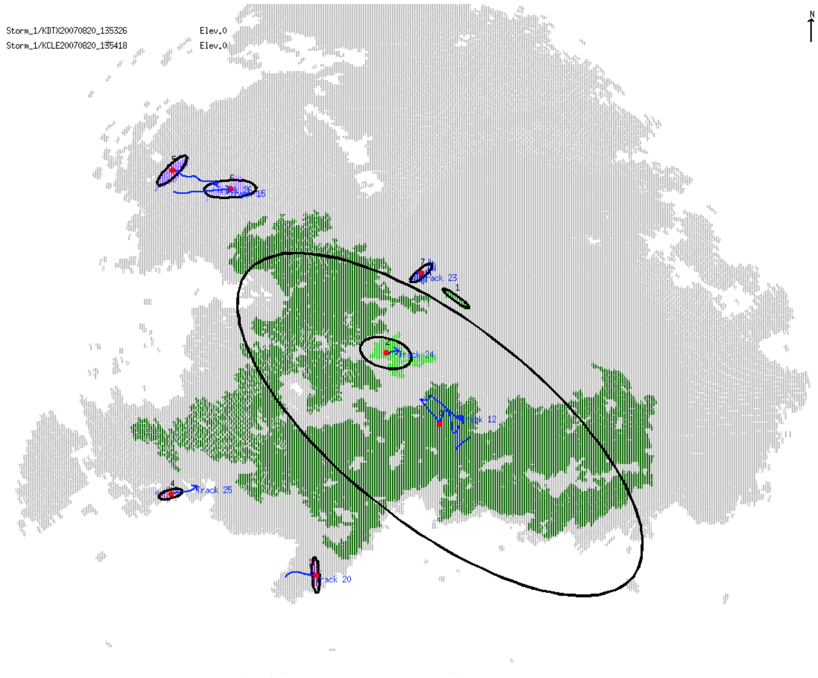
Figure D.2: The tracks on Images (ai) and (aj) of the 27 images with pseudo storms using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20^{th}, 2007



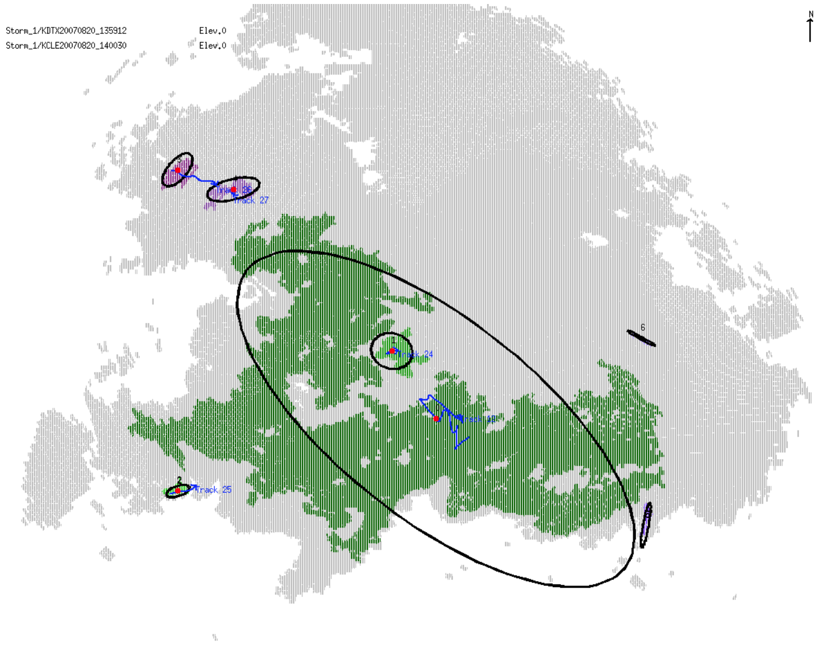
(ak)

(al)

Figure D.2: The tracks on Images (*ak*) and (*al*) of the 27 images with pseudo storms using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007

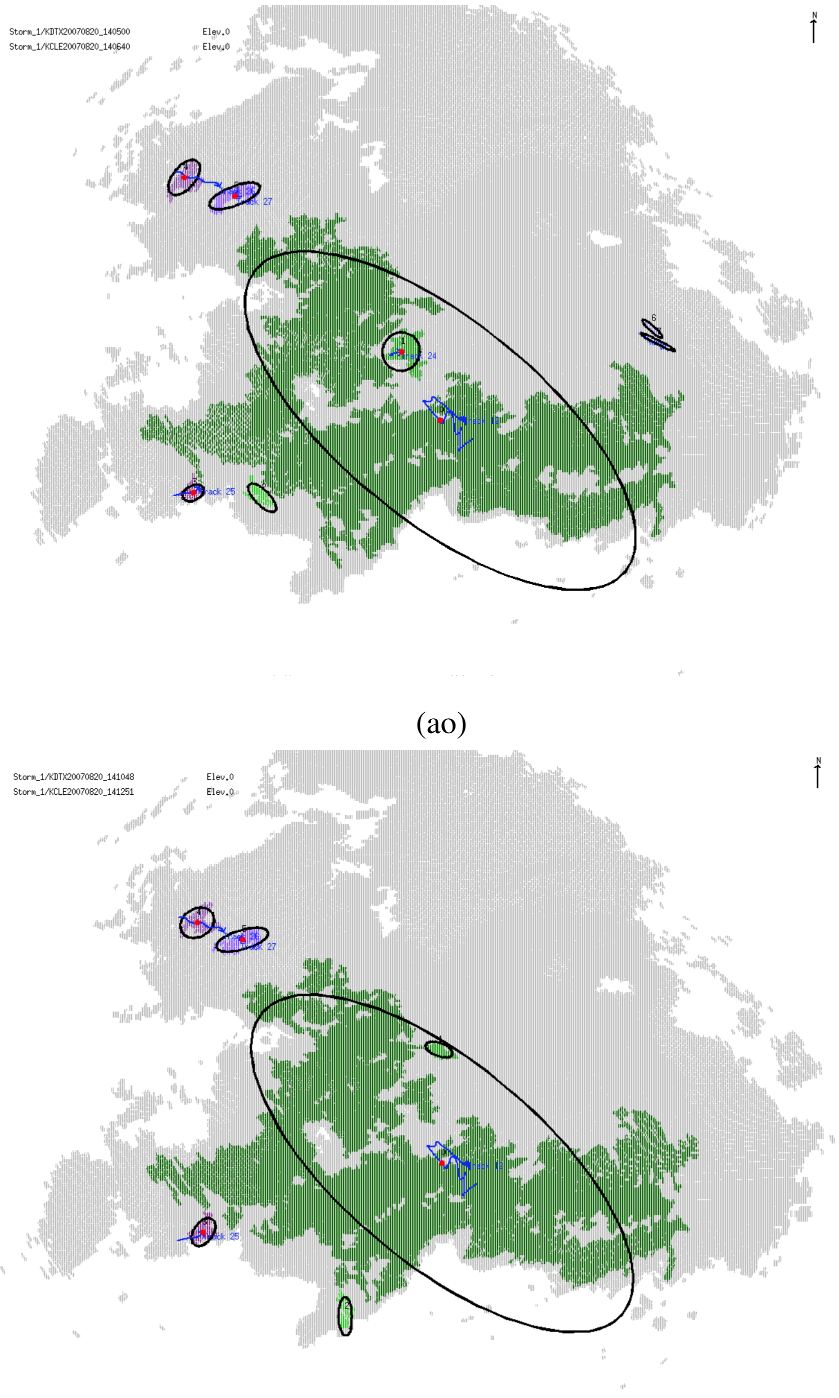


(am)



(an)

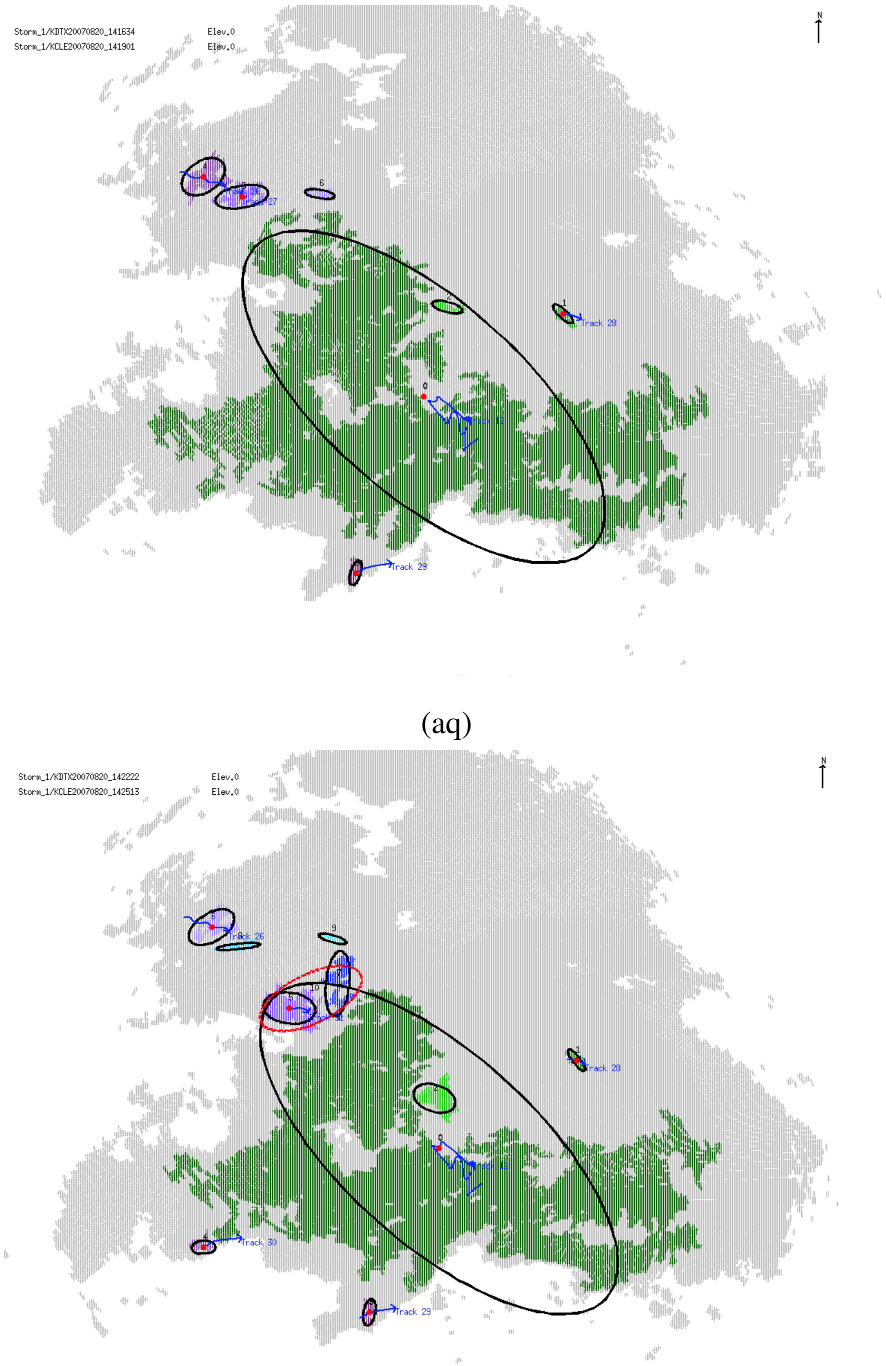
Figure D.2: The tracks on Images (*am*) and (*an*) of the 27 images with pseudo storms using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20^{th}, 2007



(ao)

(ap)

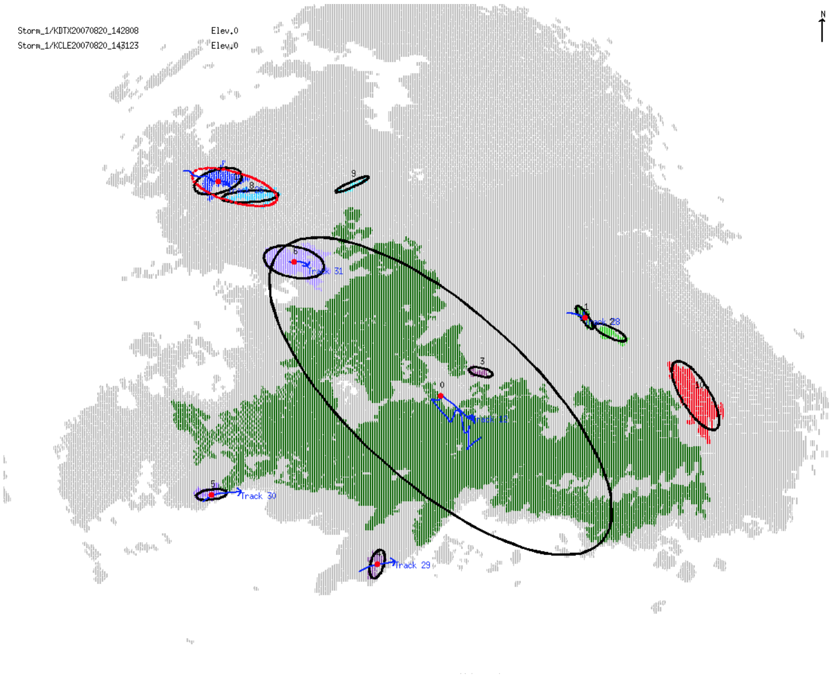
Figure D.2: The tracks on Images (*ao*) and (*ap*) of the 27 images with pseudo storms using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007



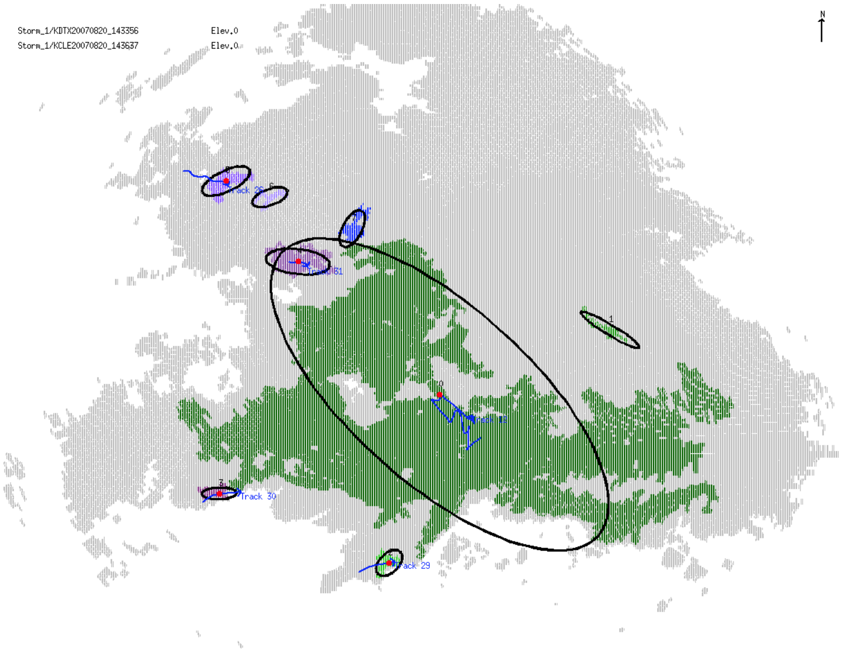
(aq)

(ar)

Figure D.2: The tracks on Images (*aq*) and (*ar*) of the 27 images with pseudo storms using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20^{th}, 2007

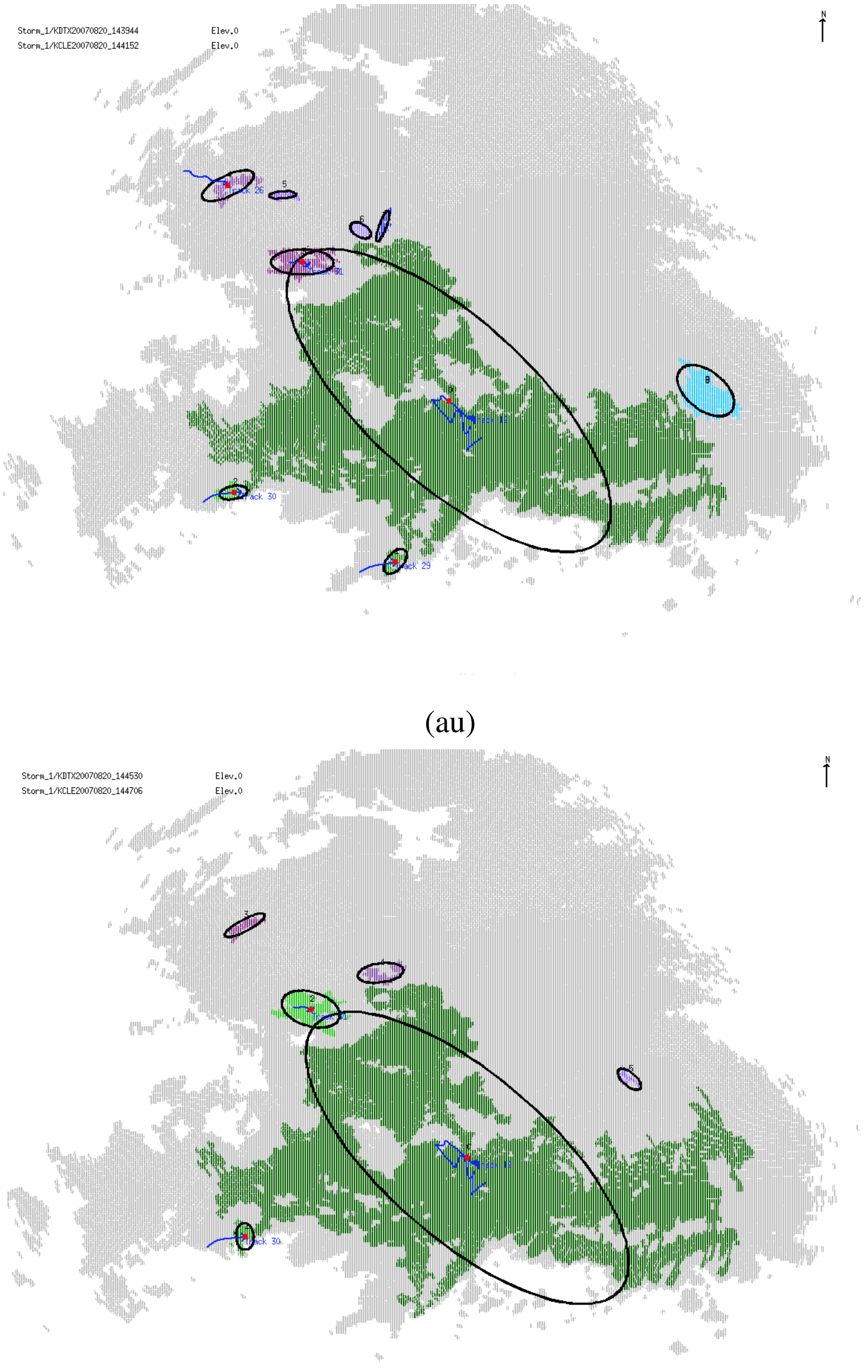


(as)



(at)

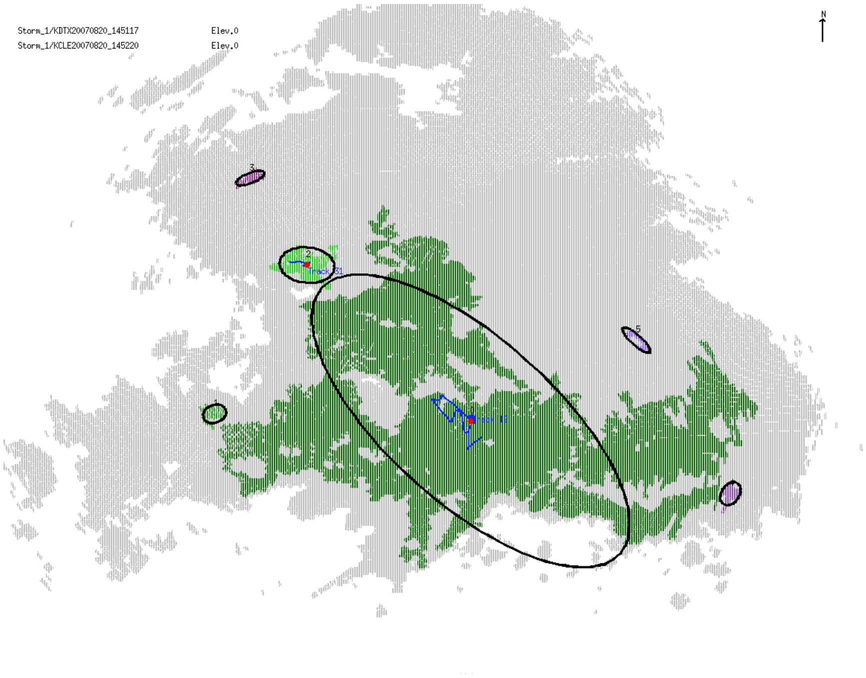
Figure D.2: The tracks on Images (*as*) and (*at*) of the 27 images with pseudo storms using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007



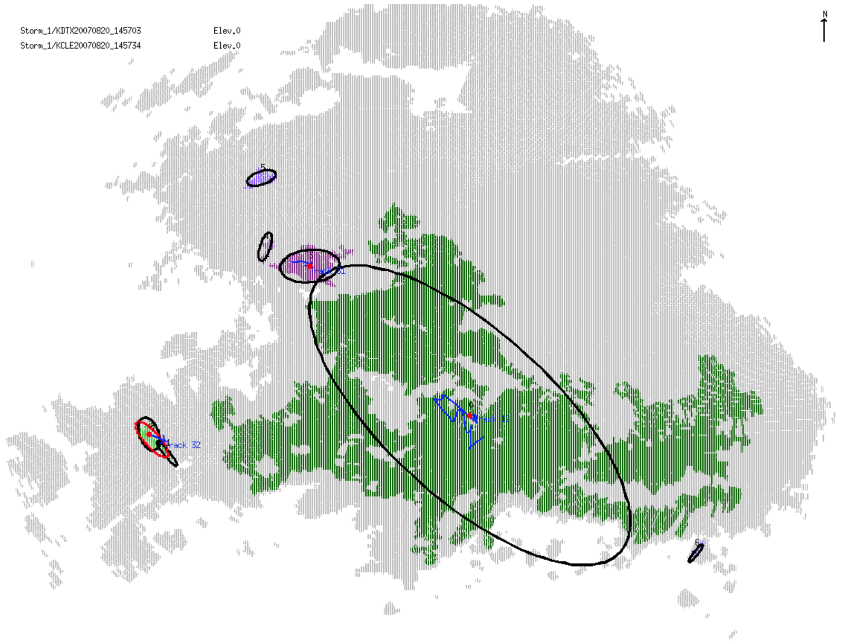
(au)

(av)

Figure D.2: The tracks on Images (*au*) and (*av*) of the 27 images with pseudo storms using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007

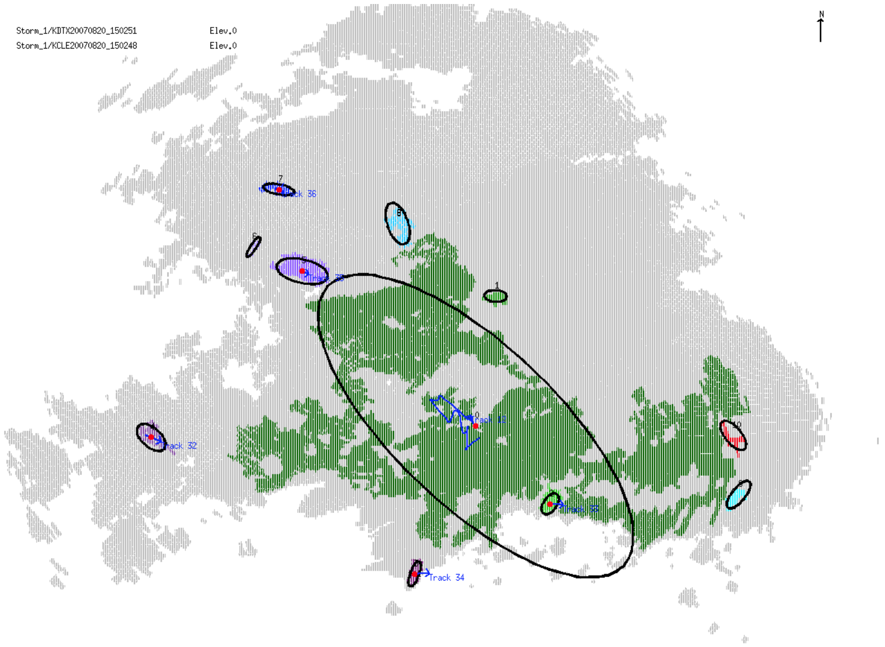


(aw)

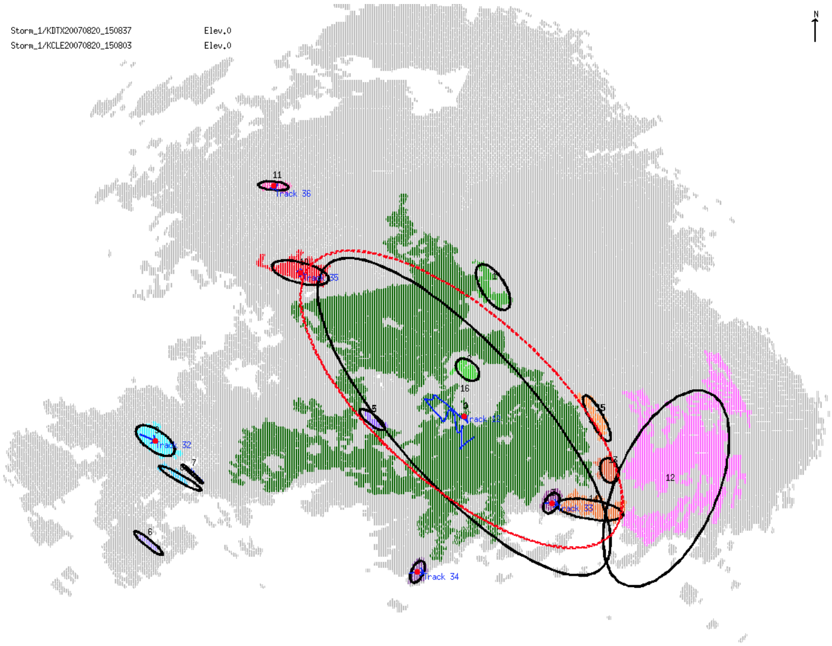


(ax)

Figure D.2: The tracks on Images (*aw*) and (*ax*) of the 27 images with pseudo storms using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20^{th}, 2007

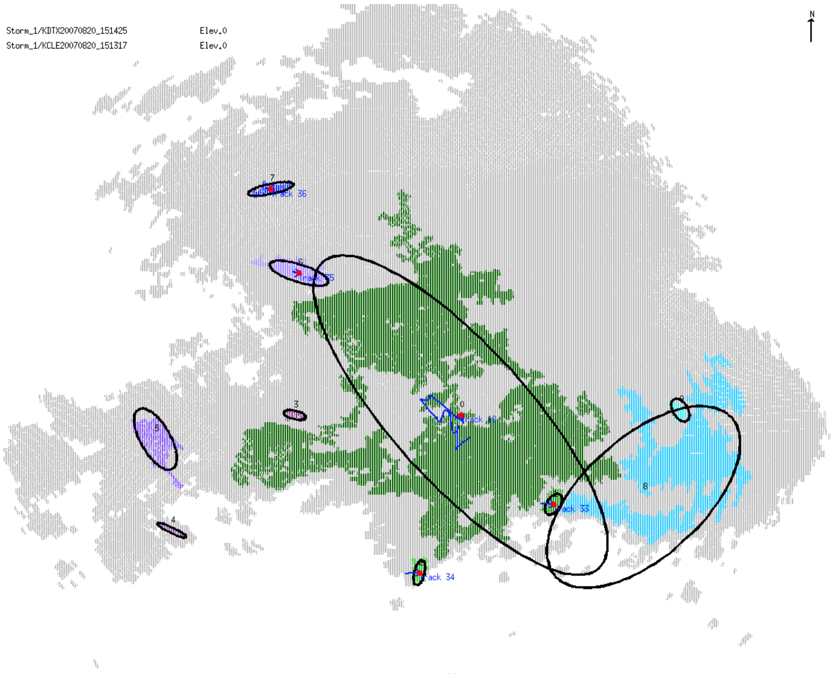


(ay)



(az)

Figure D.2: The tracks on Images (*ay*) and (*az*) of the 27 images with pseudo storms using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007

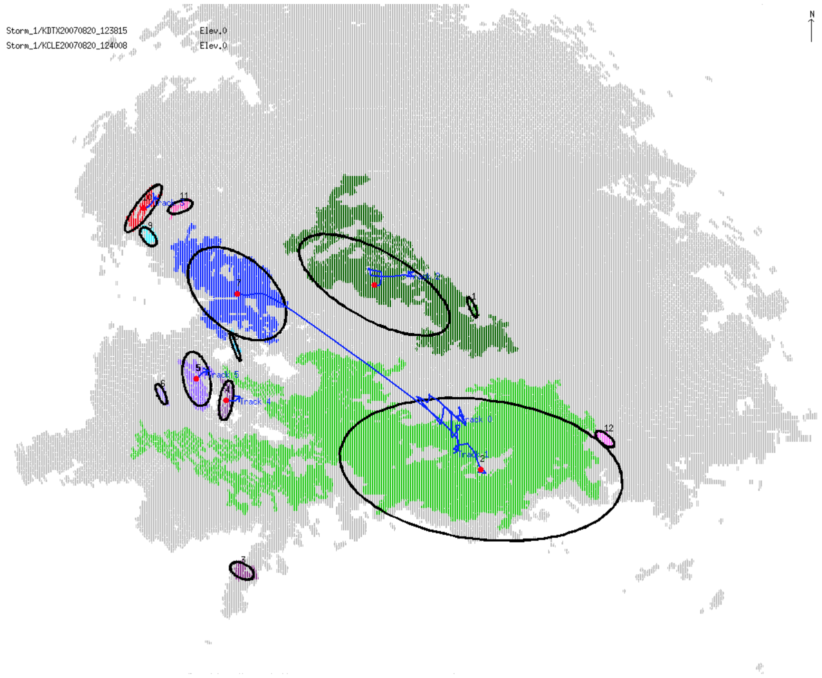


(ba)

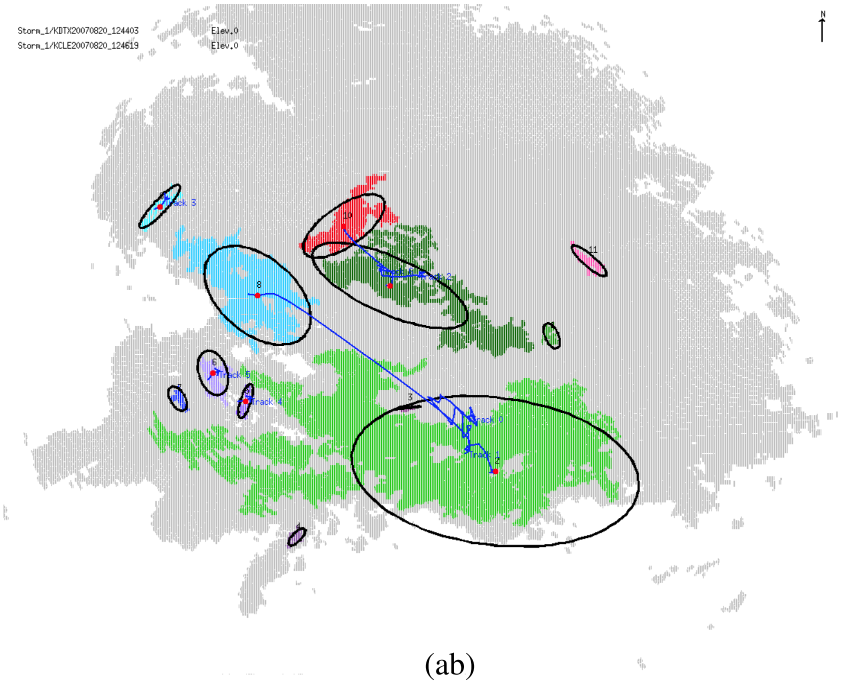
Figure D.2: The tracks on Image (*ba*) of the 27 images with pseudo storms using the original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007

D.3 Image Sequence 3

The third group of figures displays the tracking results from the Detroit and Cleveland Doppler data on August 20th, 2007 using the advanced pseudo storm relaxation labeling algorithm.

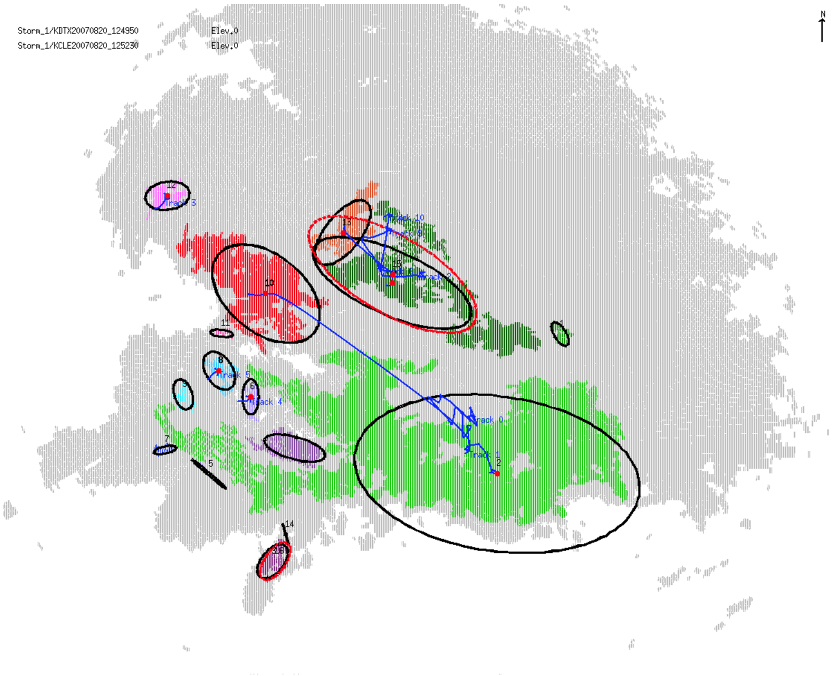


(aa)

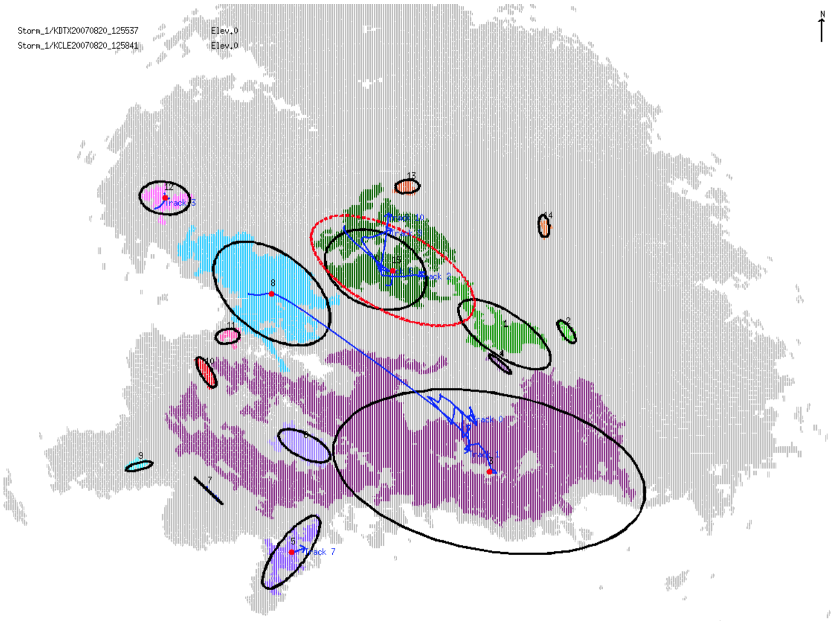


(ab)

Figure D.3: The tracks on Images (*aa*) and (*ab*) of the 27 images with pseudo storms using the pseudo relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20^{th}, 2007

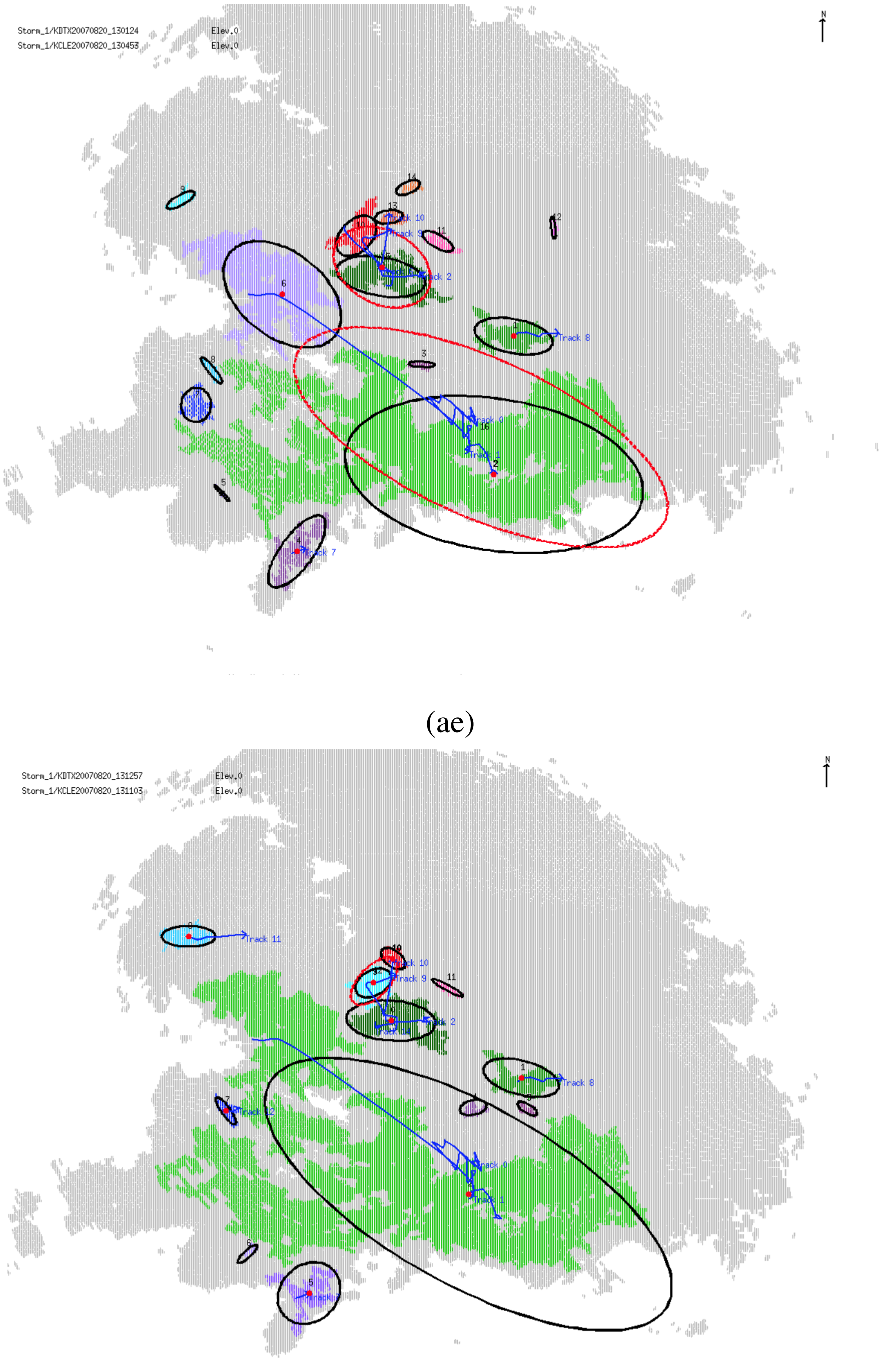


(ac)



(ad)

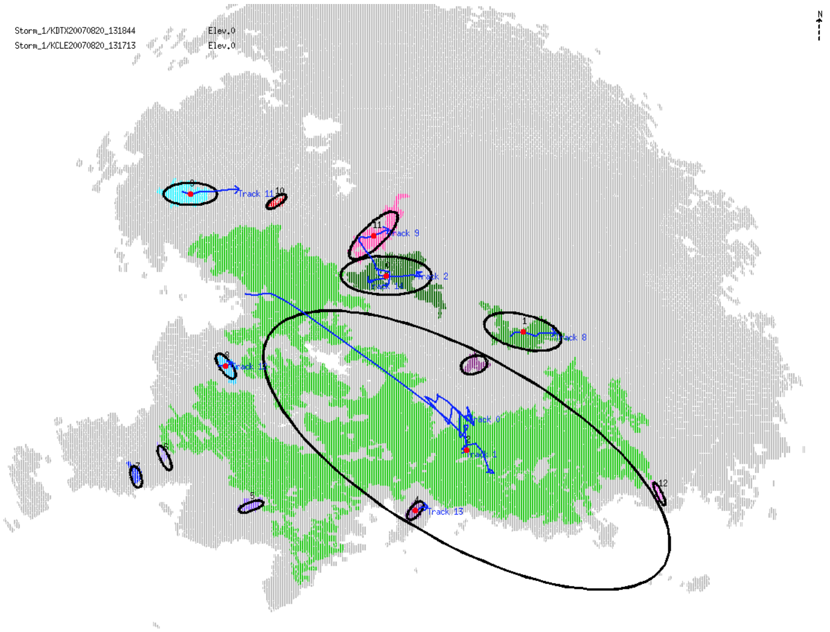
Figure D.3: The tracks on Images (*ac*) and (*ad*) of the 27 images with pseudo storms using the pseudo relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007



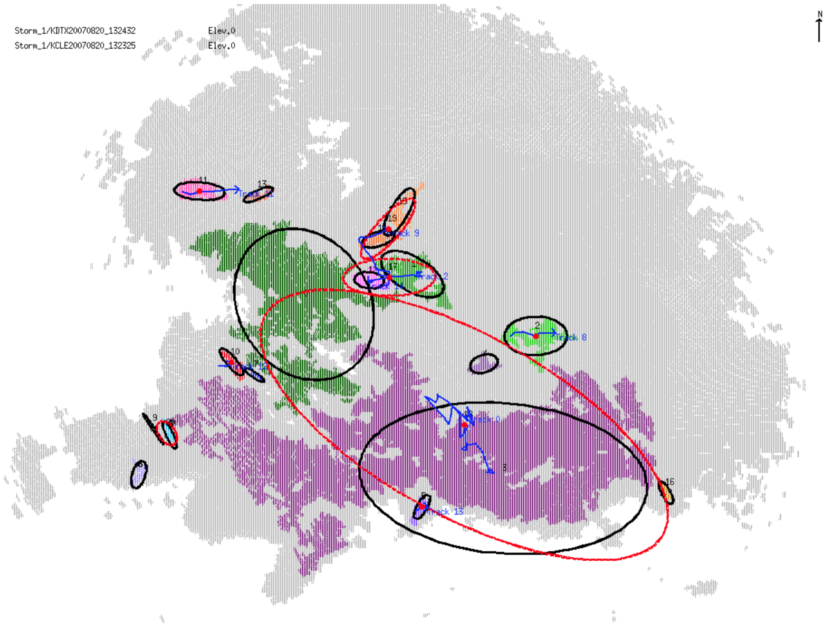
(ae)

(af)

Figure D.3: The tracks on Images (ae) and (af) of the 27 images with pseudo storms using the pseudo relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007

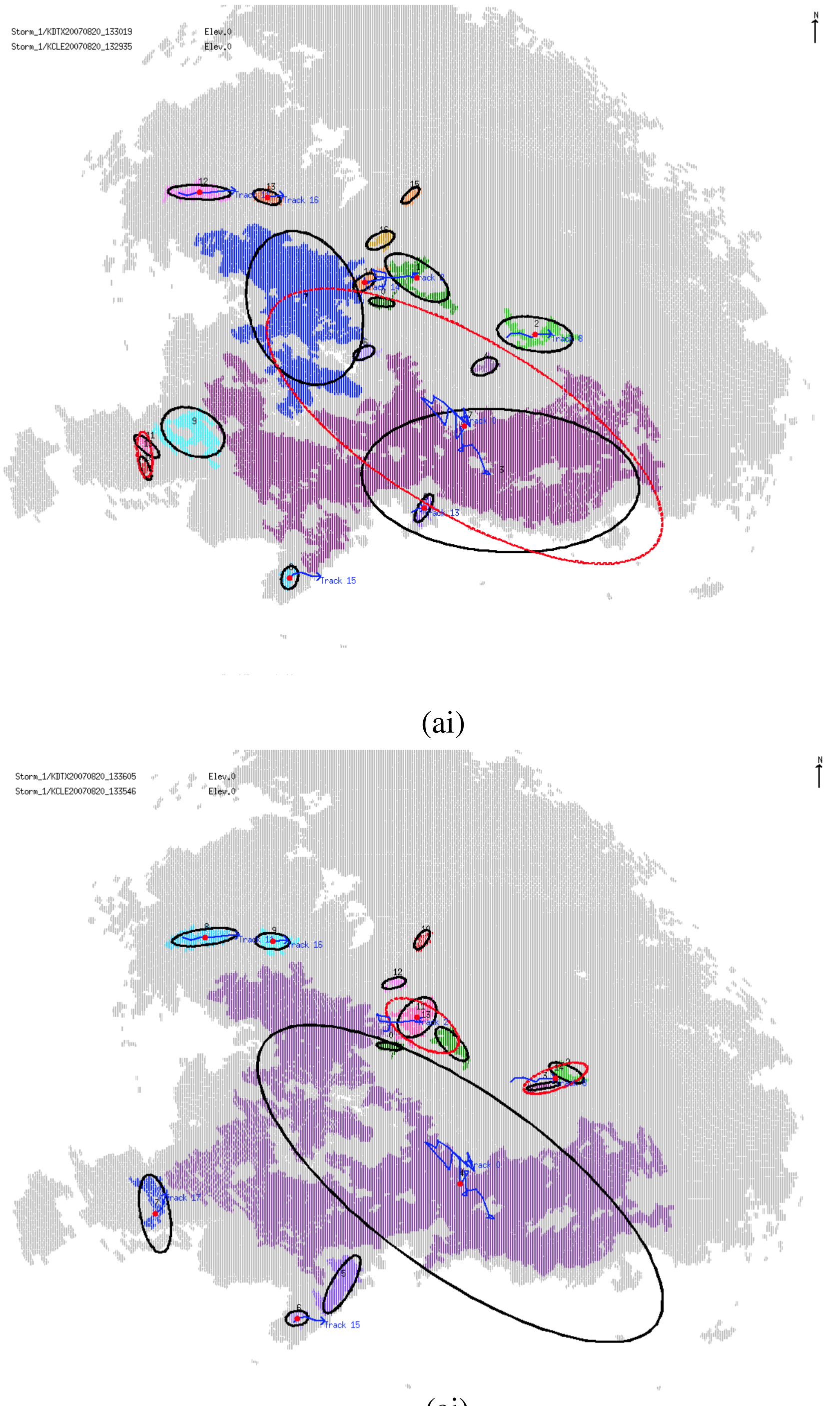


(ag)



(ah)

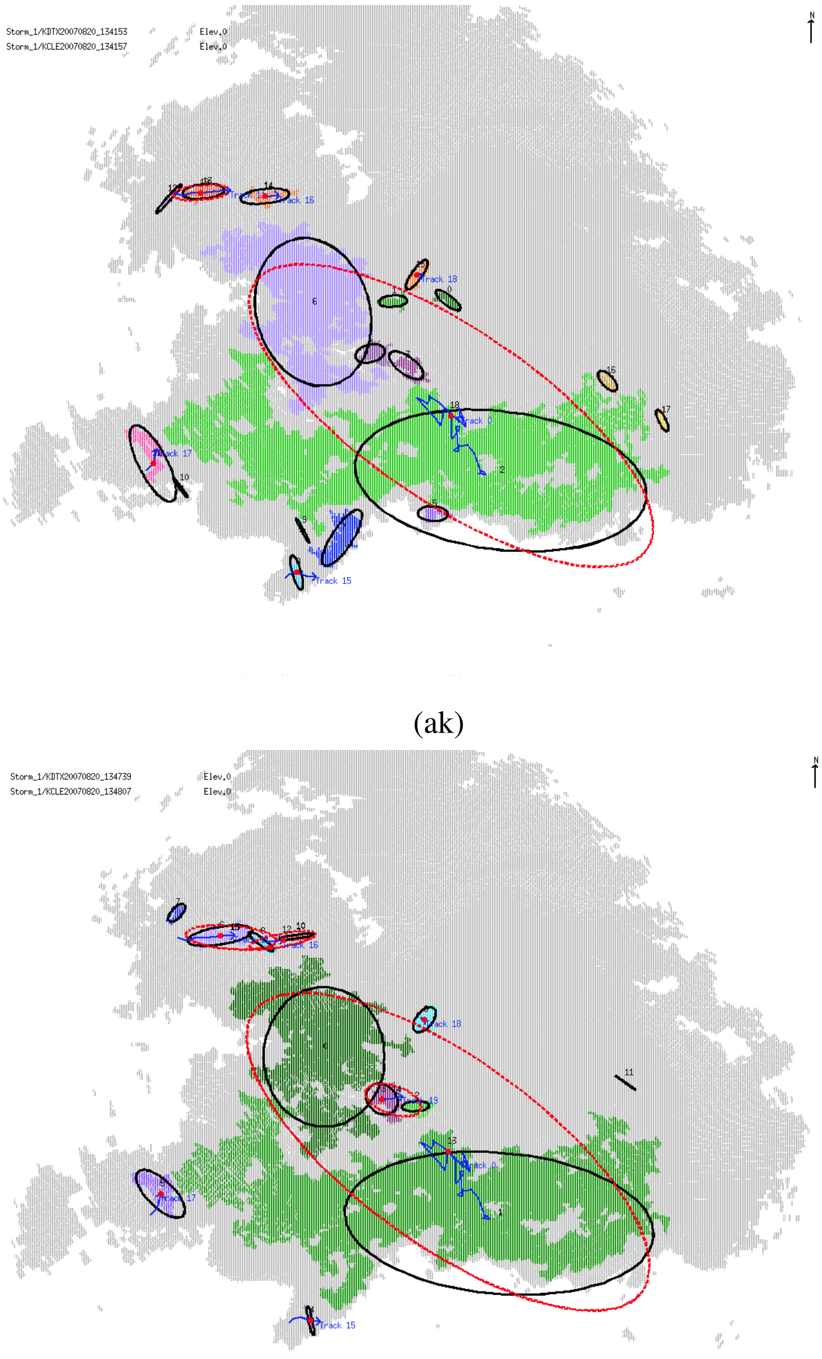
Figure D.3: The tracks on Images (*ag*) and (*ah*) of the 27 images with pseudo storms using the pseudo relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007



(ai)

(aj)

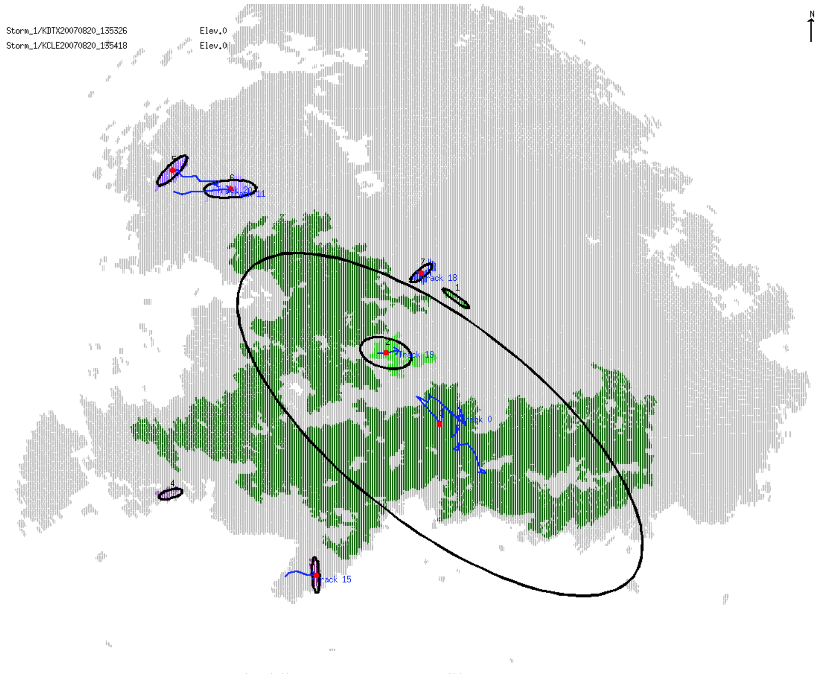
Figure D.3: The tracks on Images (ai) and (aj) of the 27 images with pseudo storms using the pseudo relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007



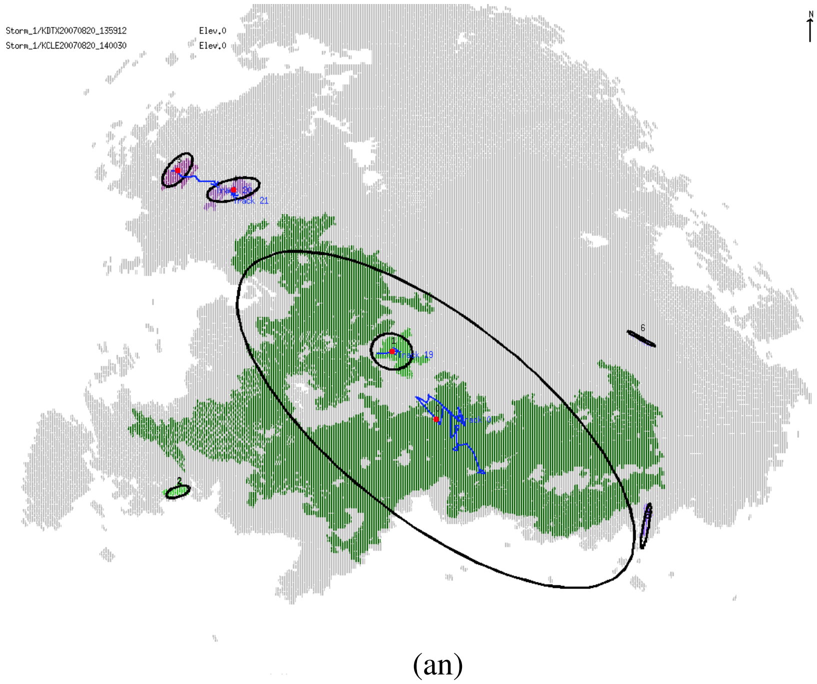
(ak)

(al)

Figure D.3: The tracks on Images (*ak*) and (*al*) of the 27 images with pseudo storms using the pseudo relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007

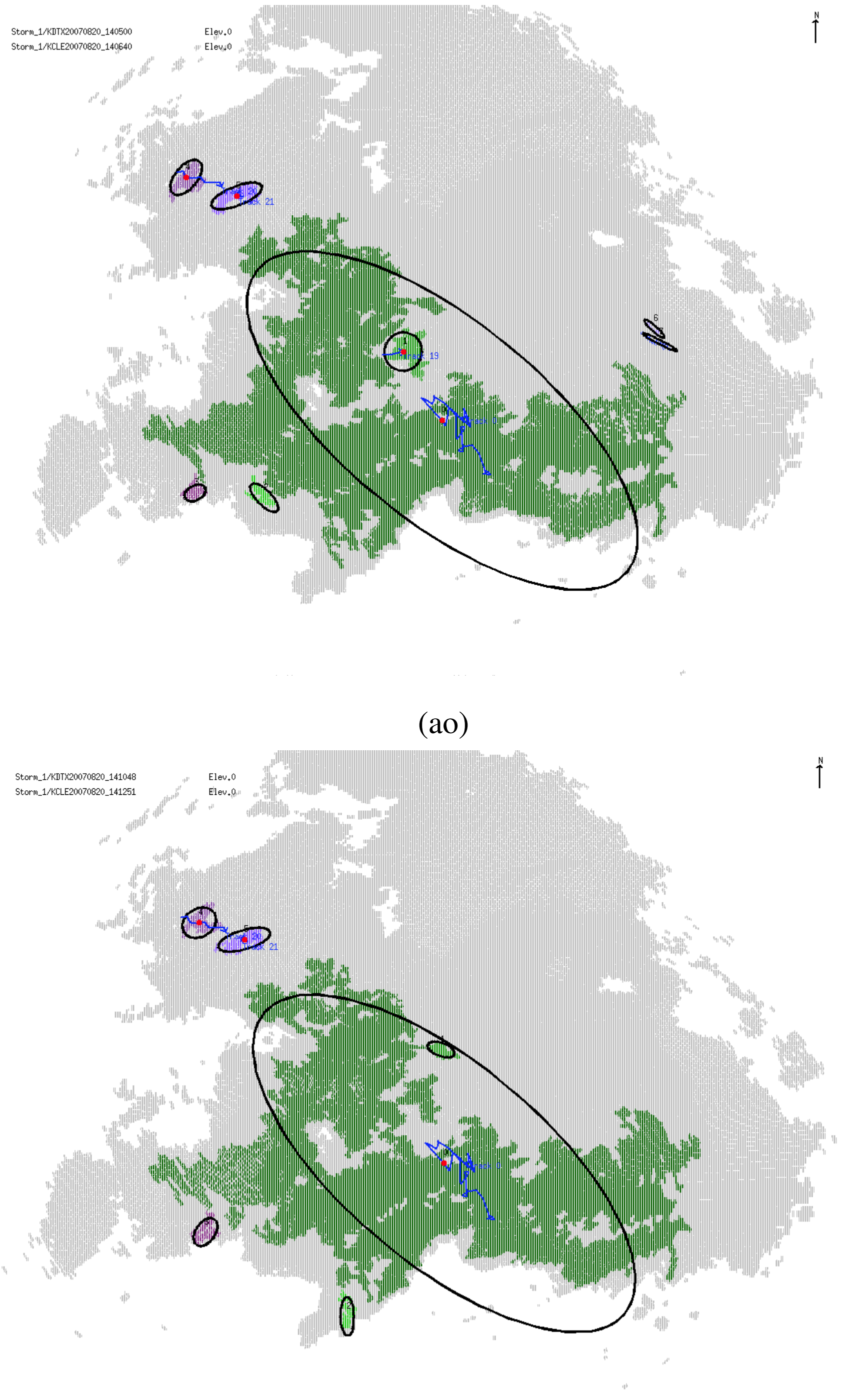


(am)



(an)

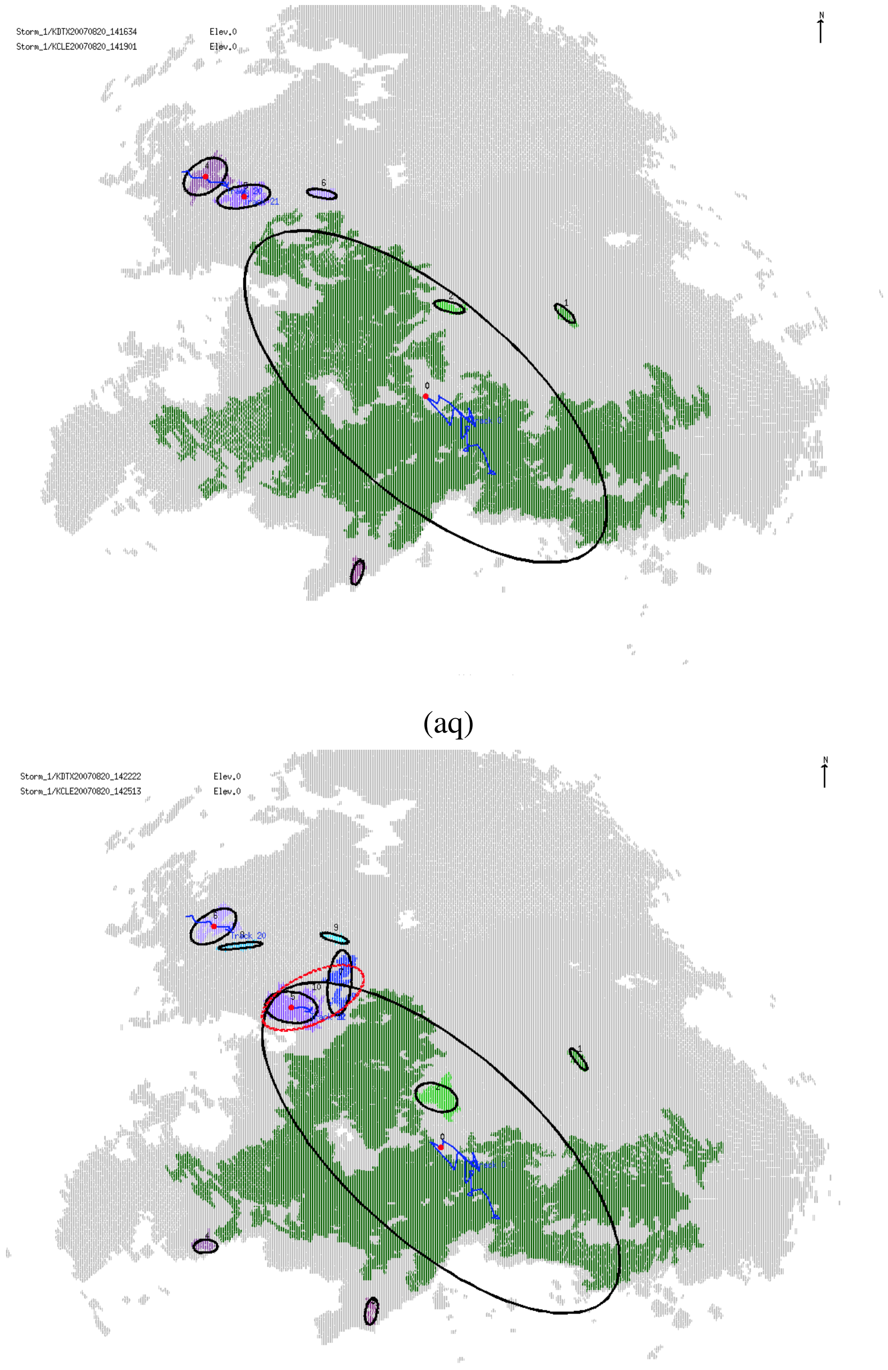
Figure D.3: The tracks on Images (*am*) and (*an*) of the 27 images with pseudo storms using the pseudo relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20^{th}, 2007



(ao)

(ap)

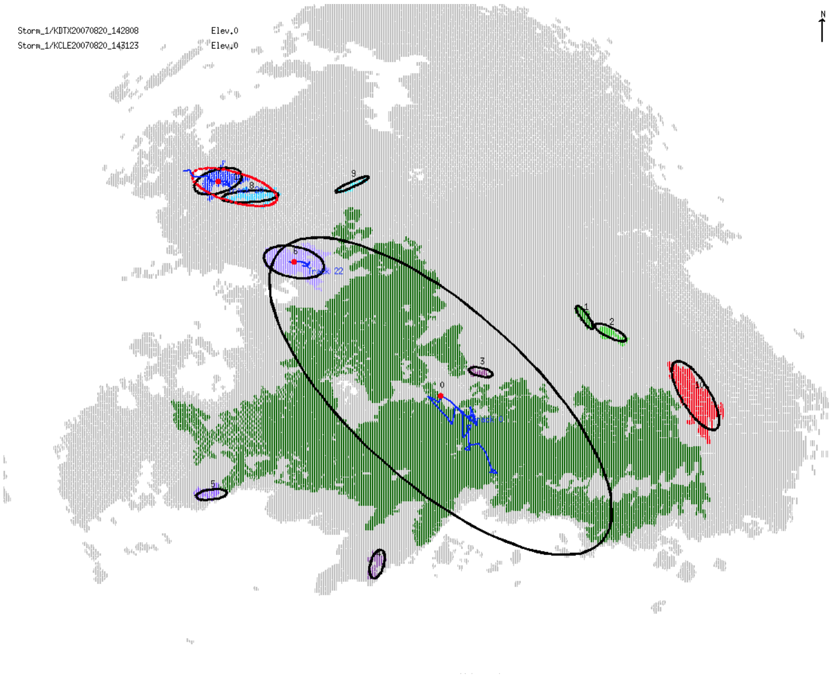
Figure D.3: The tracks on Images (*ao*) and (*ap*) of the 27 images with pseudo storms using the pseudo relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007



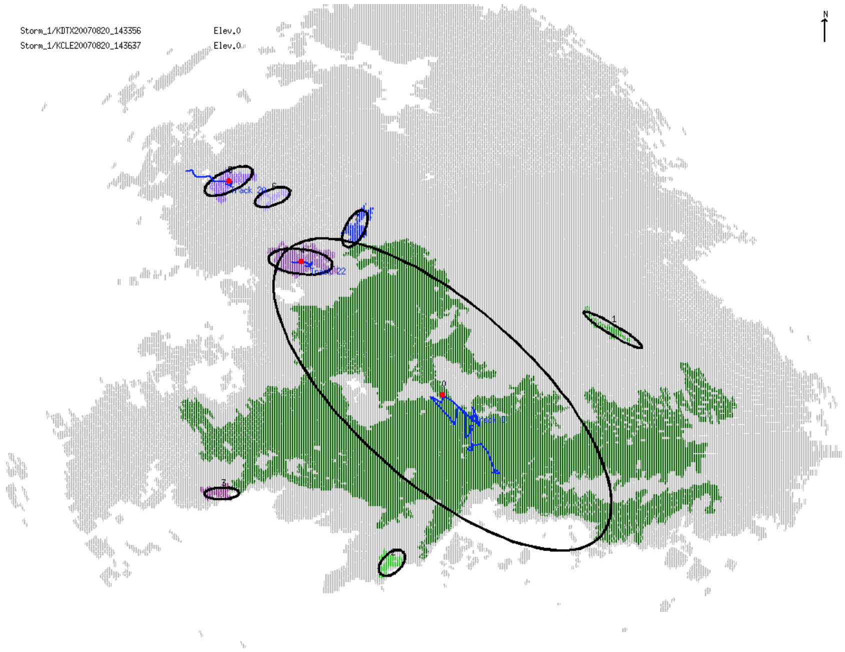
(aq)

(ar)

Figure D.3: The tracks on Images (*aq*) and (*ar*) of the 27 images with pseudo storms using the pseudo relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007

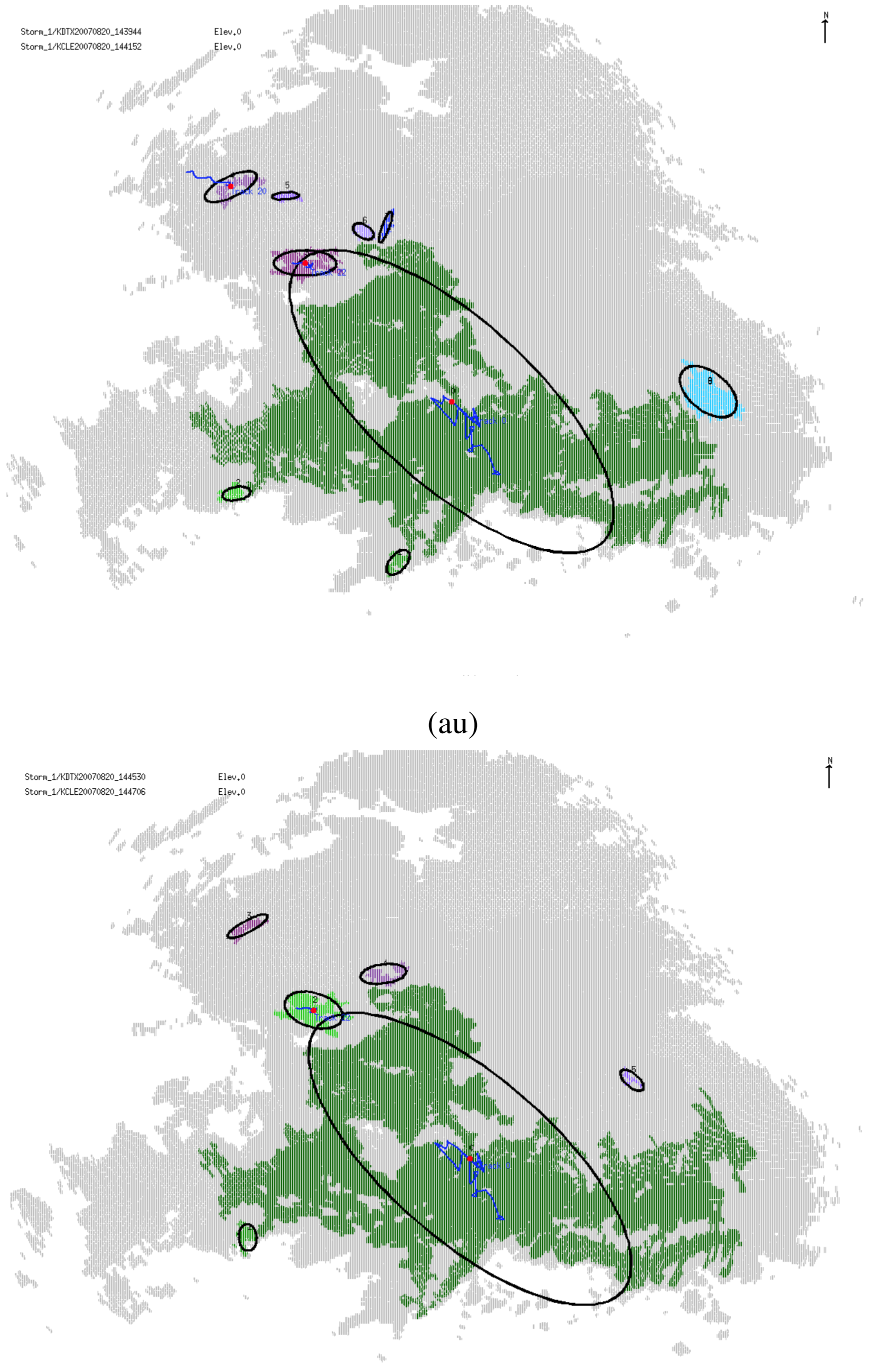


(as)



(at)

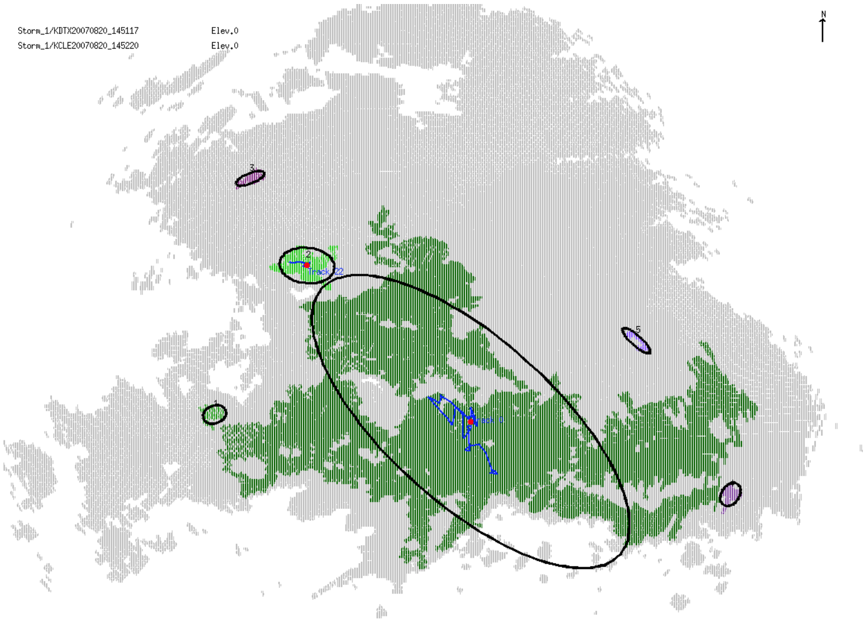
Figure D.3: The tracks on Images (as) and (at) of the 27 images with pseudo storms using the pseudo relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007



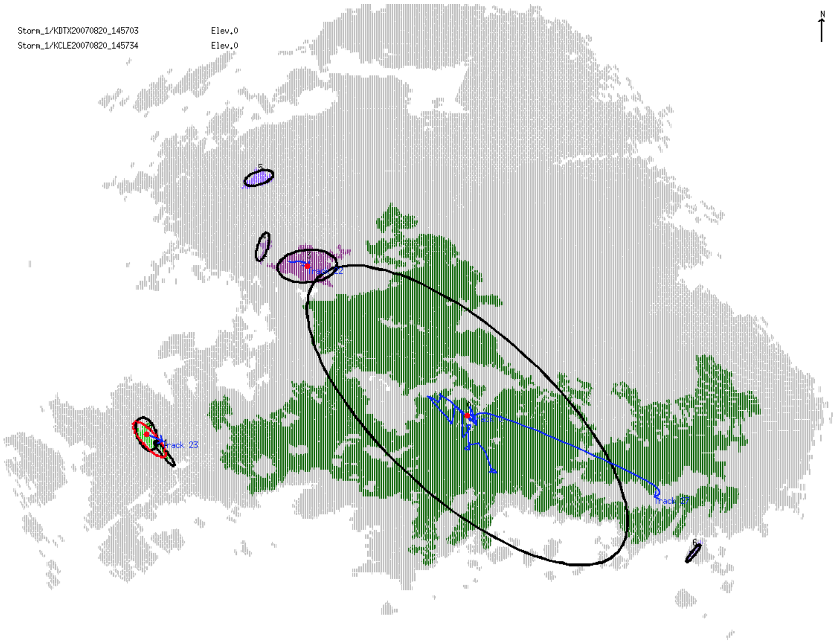
(au)

(av)

Figure D.3: The tracks on Images (*au*) and (*av*) of the 27 images with pseudo storms using the pseudo relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007

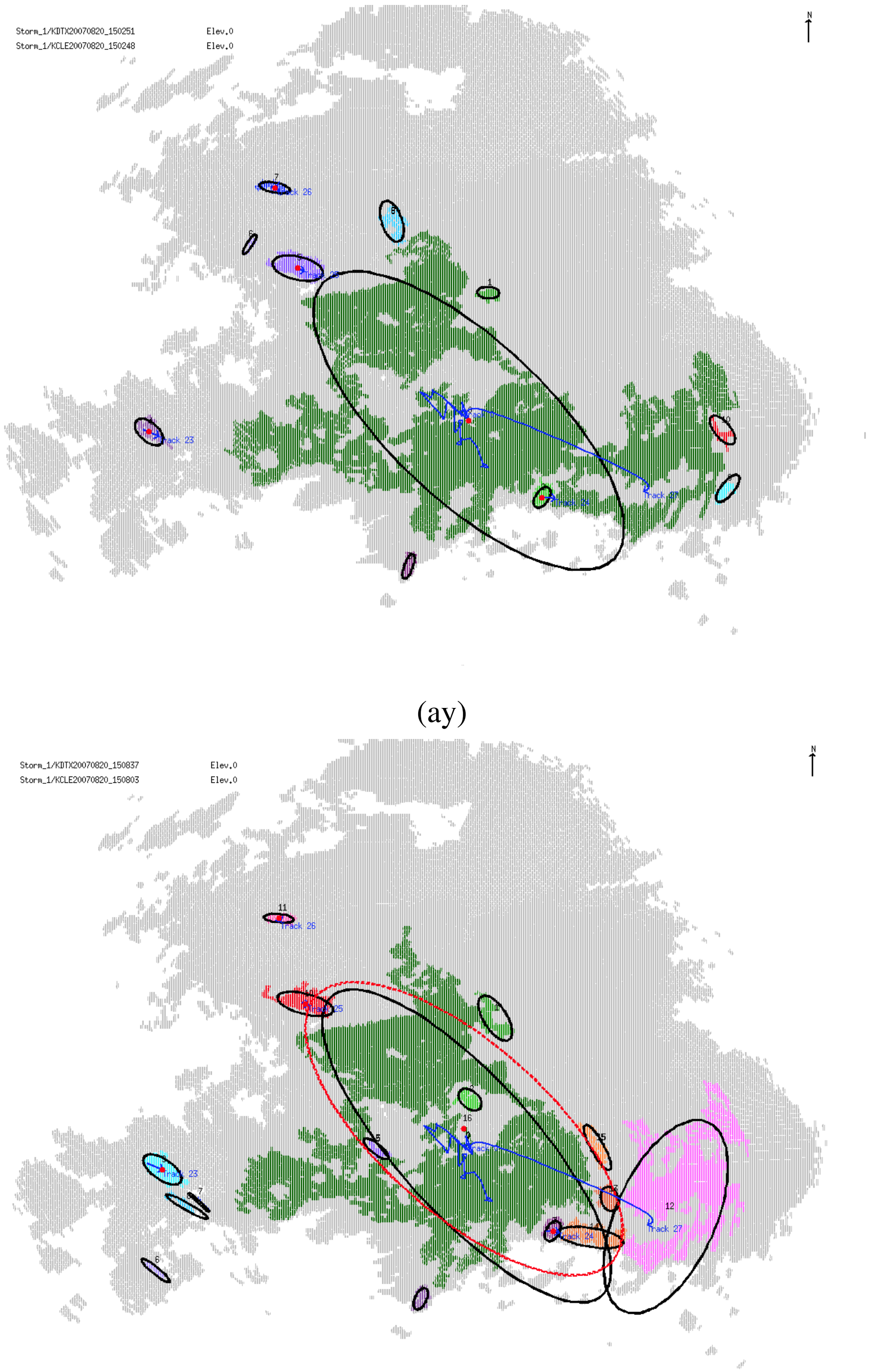


(aw)



(ax)

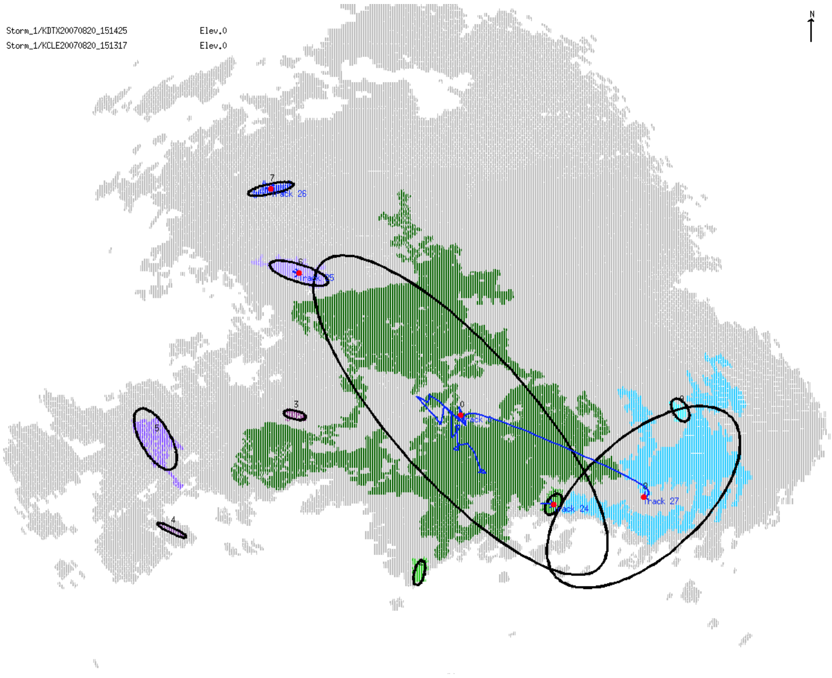
Figure D.3: The tracks on Images (*aw*) and (*ax*) of the 27 images with pseudo storms using the pseudo relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007



(ay)

(az)

Figure D.3: The tracks on Images (*ay*) and (*az*) of the 27 images with pseudo storms using the pseudo relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20th, 2007



(ba)

Figure D.3: The tracks on Image (*ba*) of the 27 images with pseudo storms using the pseudo relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 20^{th}, 2007

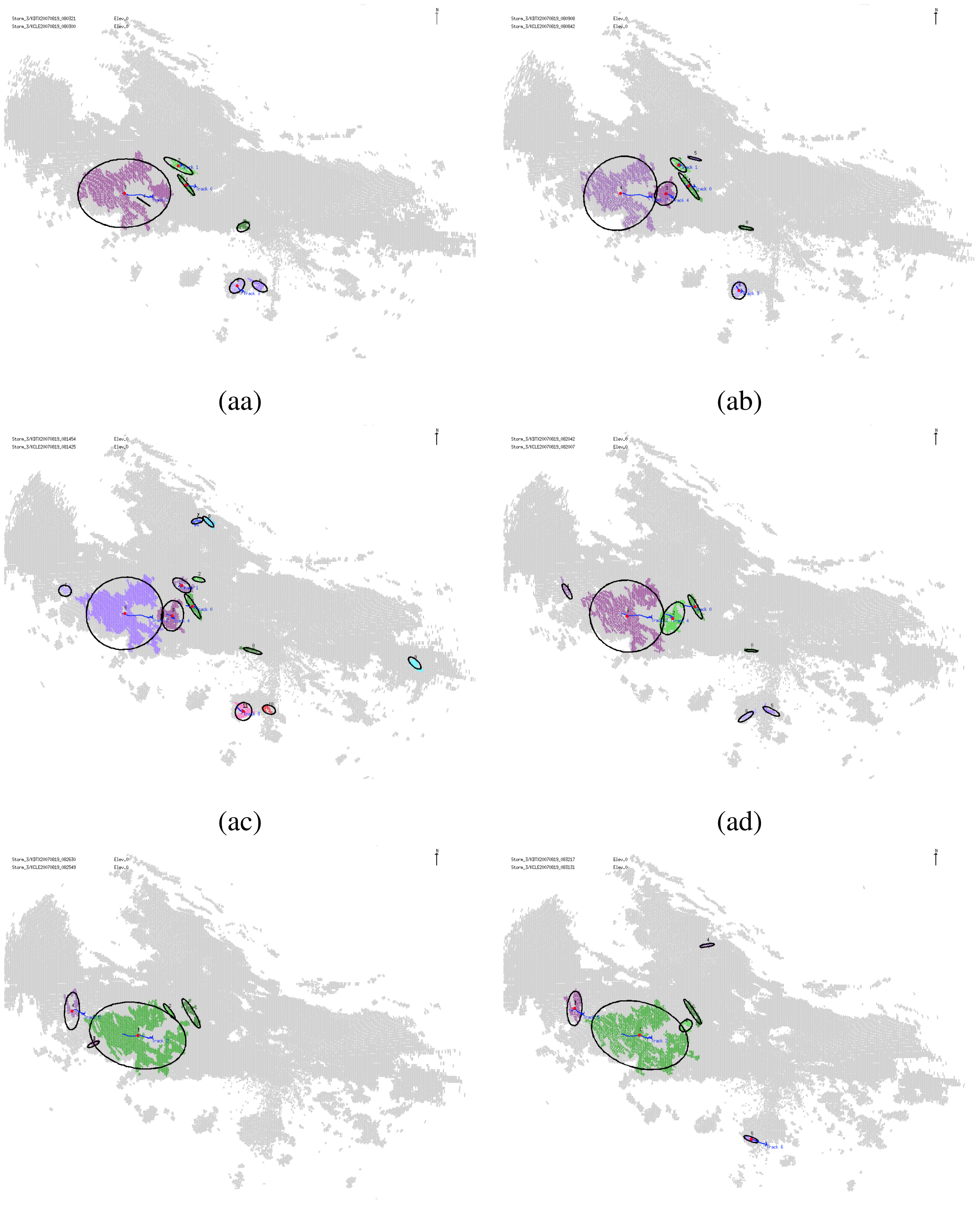
Appendix E

Tracking Results with Multiple Doppler Radars on August 19^{th}, 2007

E.1 Image Sequence 1

The first group of figures displays the tracking results from the Detroit and Cleveland Doppler data on August 19^{th}, 2007 using the original relaxation labeling algorithm.

In each image, the large grey-coloured area indicates non-zero dBZ but no storm. All of the other colours show different recognized storms. The ellipses drawn on the storms represent the size, location of the center and orientation of the ellipsoids representing the storms. The blue curves show the complete tracks of the storms and the red dots on the tracks show the current positions of the storms on the tracks.

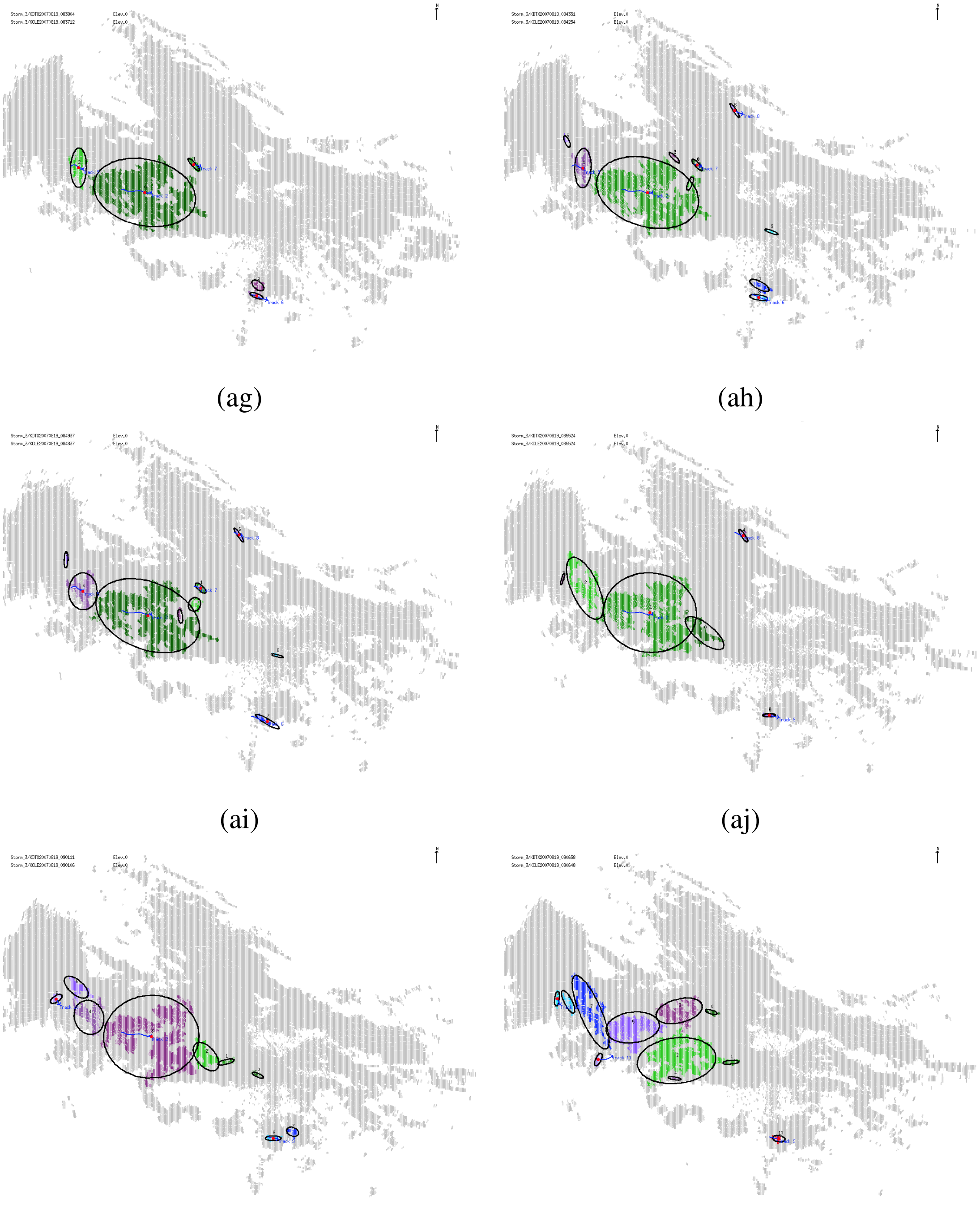


(aa) (ab)

(ac) (ad)

(ae) (af)

Figure E.1: The tracks on Images (aa), (ab), (ac), (ad), (ae) and (af) of the 64 images using original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19^{th}, 2007

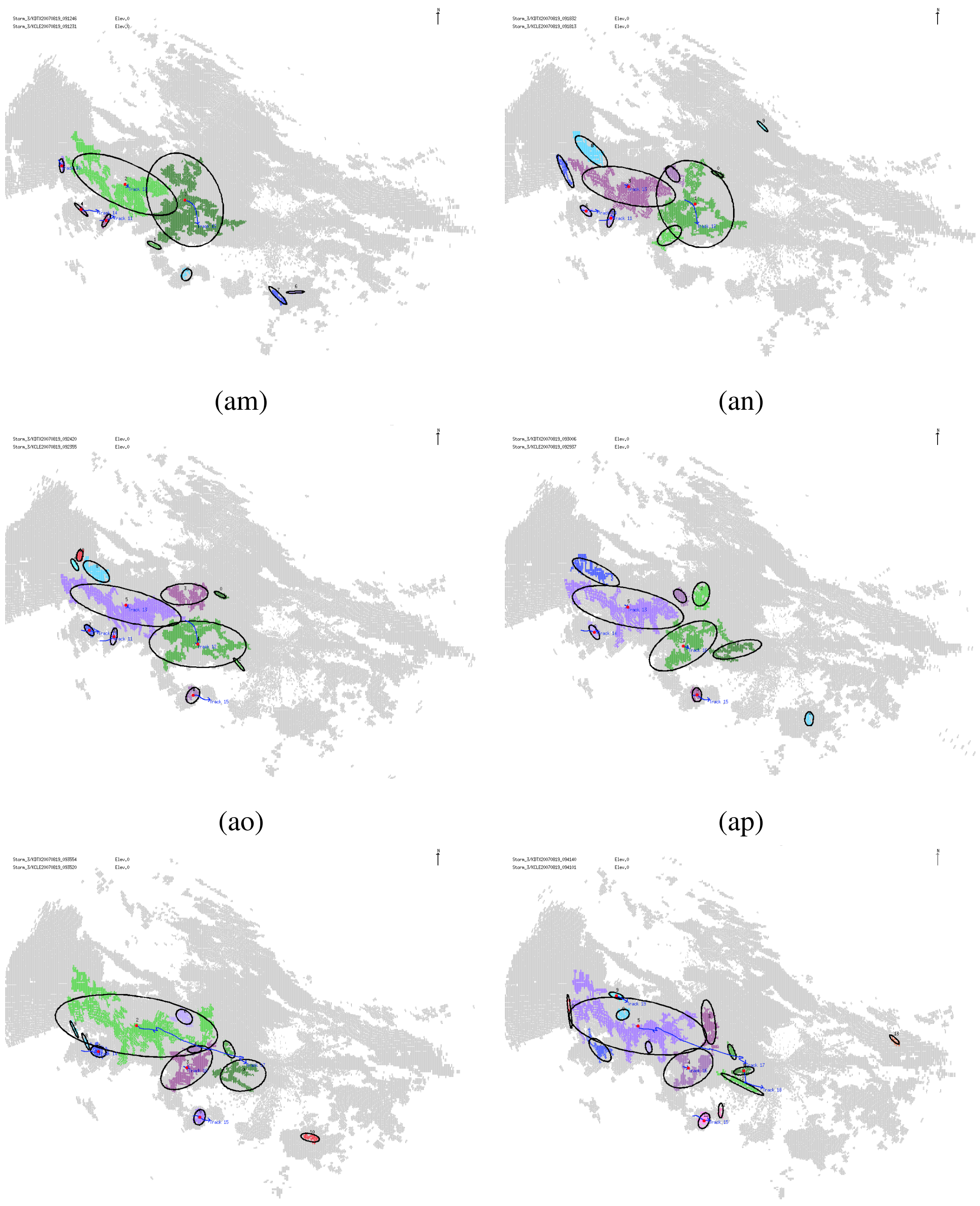


(ag) (ah)

(ai) (aj)

(ak) (al)

Figure E.2: The tracks on Images (ag), (ah), (ai), (aj), (ak) and (al) of the 64 images using original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19^{th}, 2007

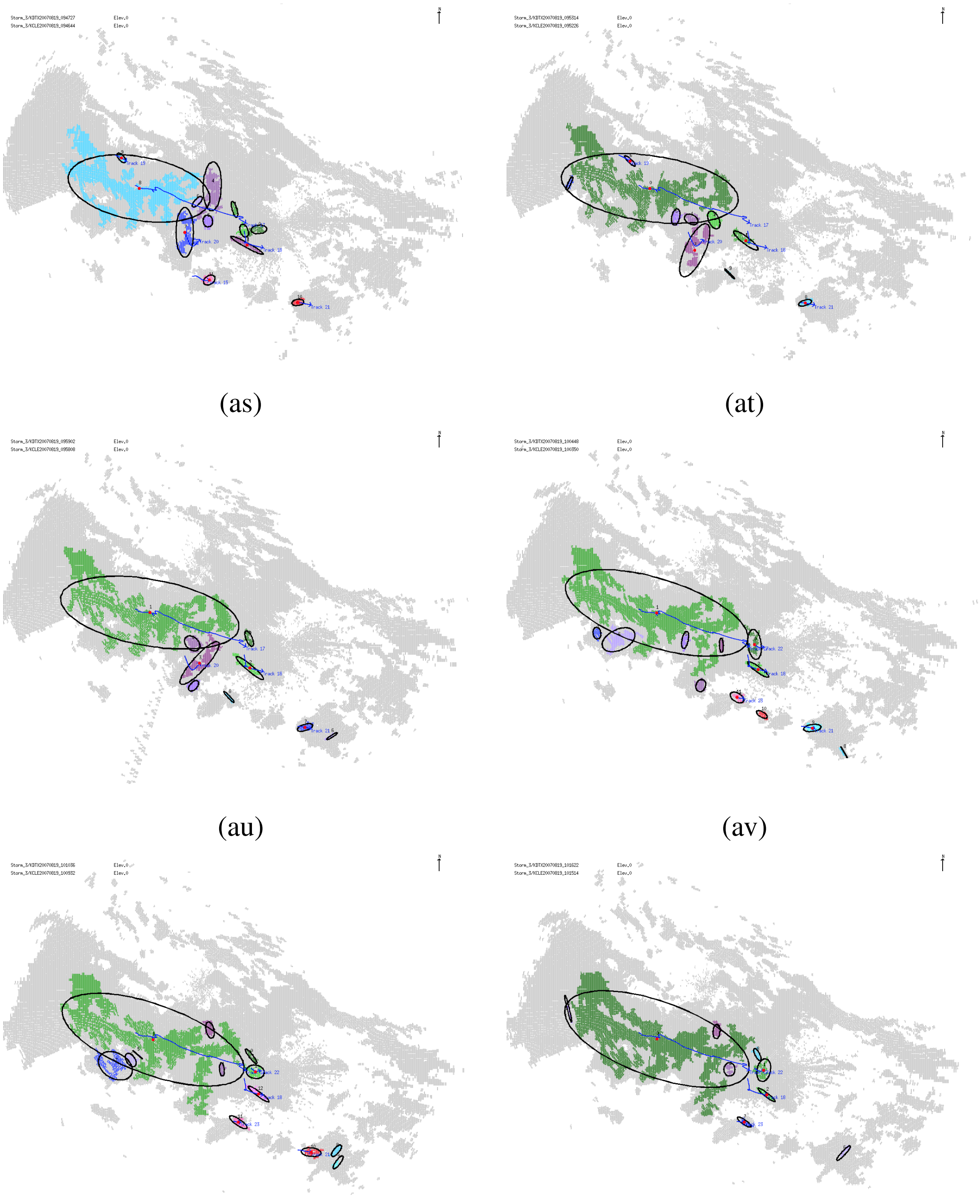


(am) (an)

(ao) (ap)

(aq) (ar)

Figure E.3: The tracks on Images (*am*), (*an*), (*ao*), (*ap*), (*aq*) and (*ar*) of the 64 images using original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19th, 2007

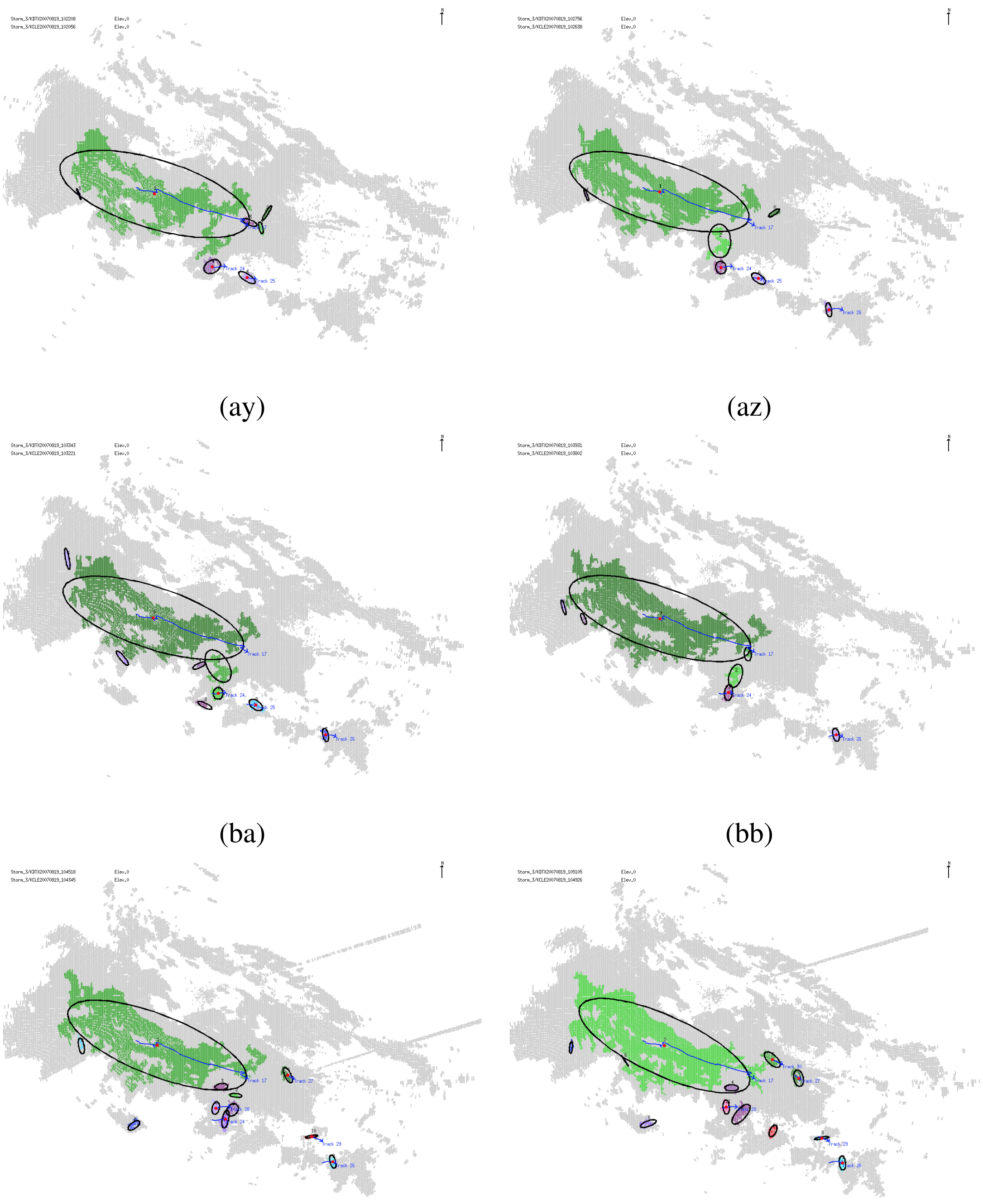


(as) (at)

(au) (av)

(aw) (ax)

Figure E.4: The tracks on Images (as), (at), (au), (av), (aw) and (ax) of the 64 images using original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19th, 2007

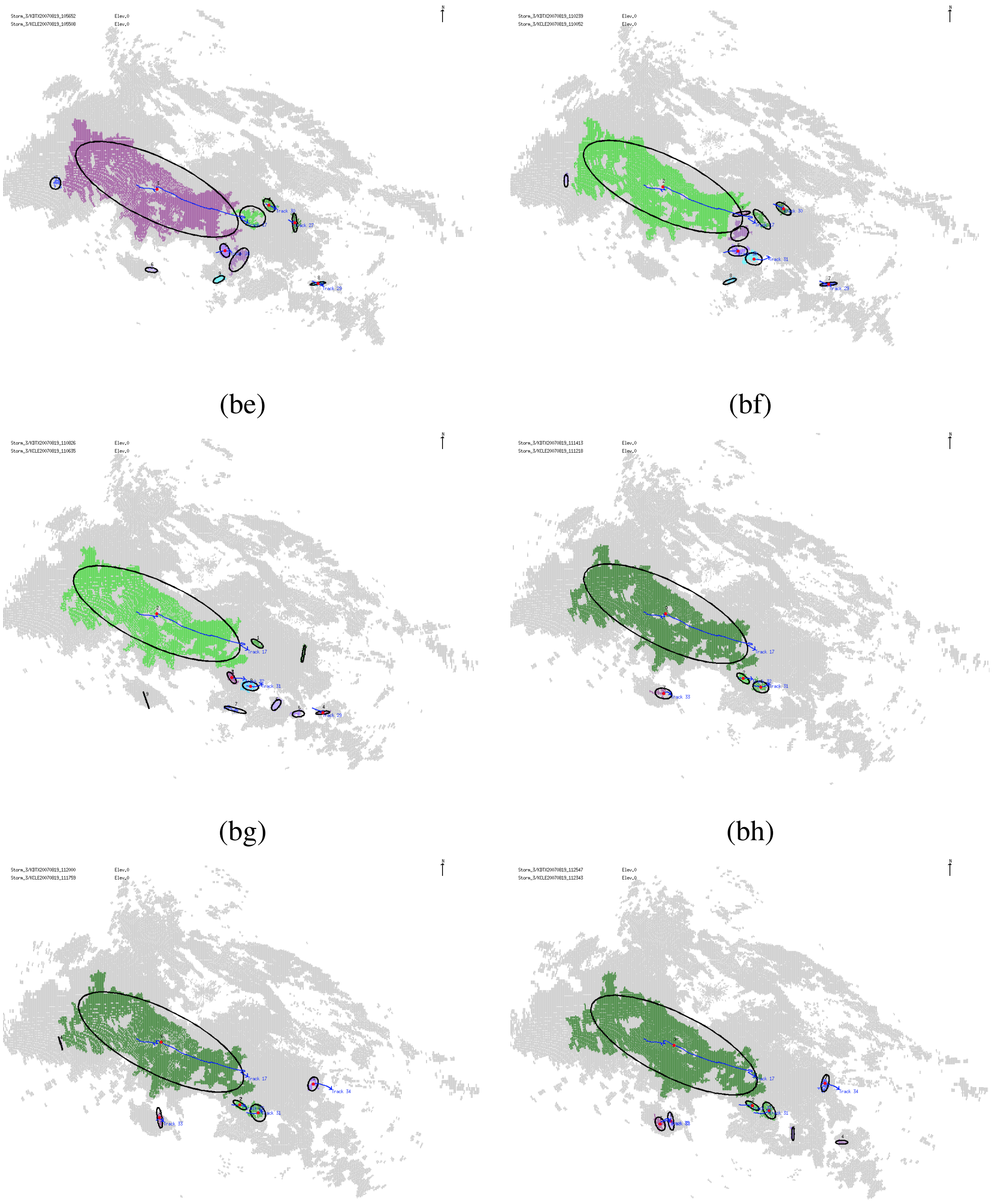


(ay) (az)

(ba) (bb)

(bc) (bd)

Figure E.5: The tracks on Images (*ay*), (*az*), (*ba*), (*bb*), (*bc*) and (*bd*) of the 64 images using original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19^{th}, 2007

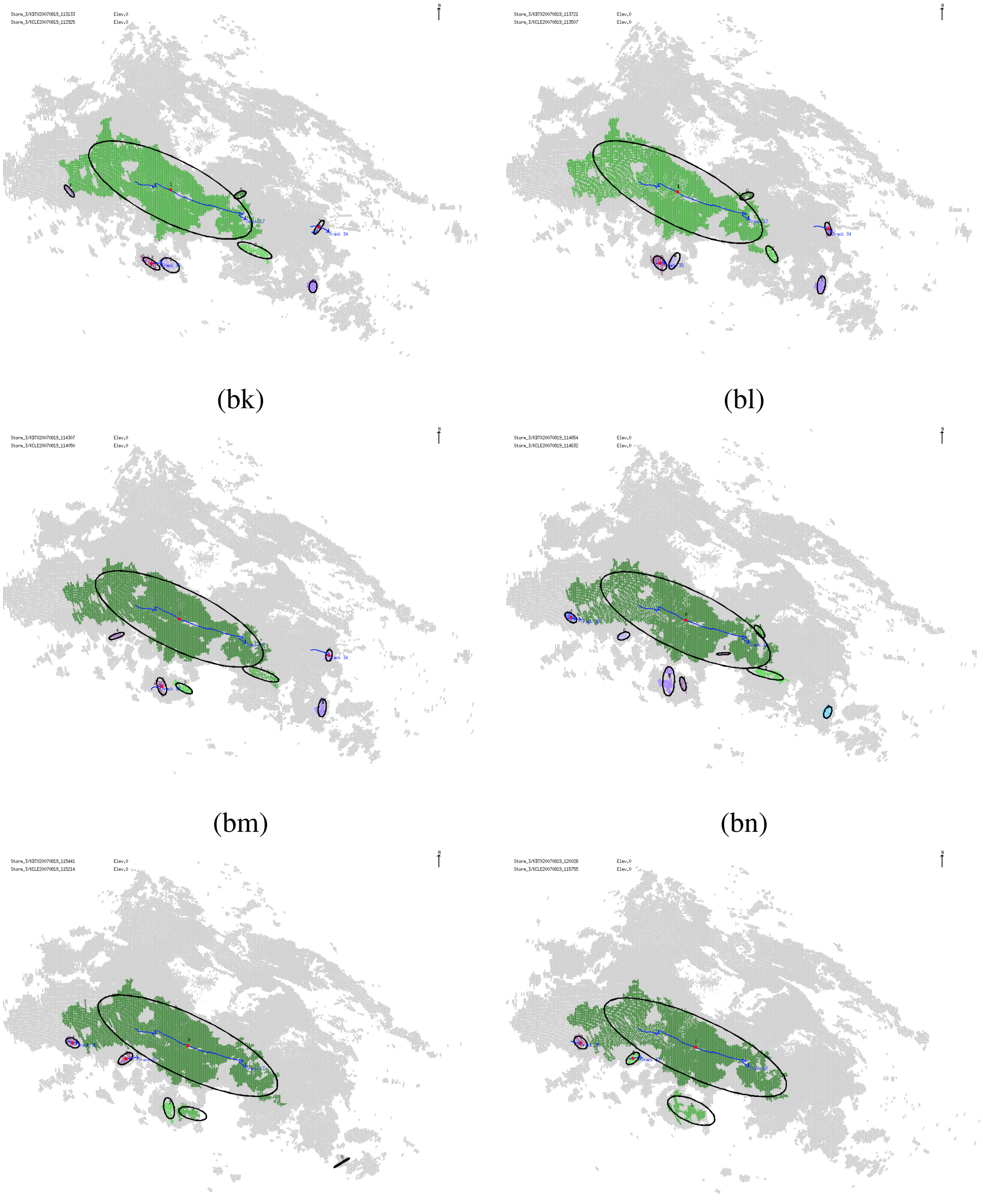


(be) (bf)

(bg) (bh)

(bi) (bj)

Figure E.6: The tracks on Images (be), (bf), (bg), (bh), (bi) and (bj) of the 64 images using original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19th, 2007

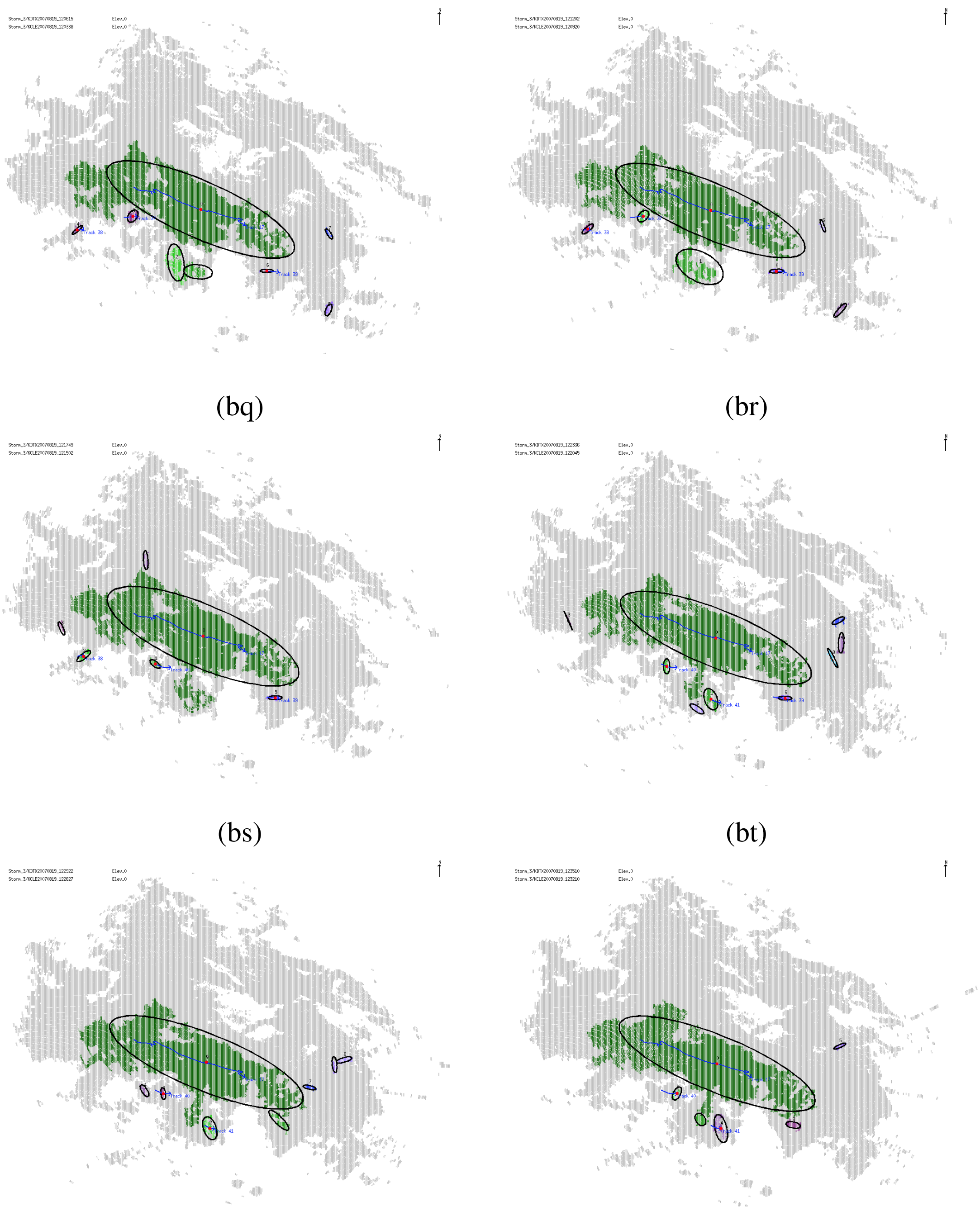


(bk) (bl)

(bm) (bn)

(bo) (bp)

Figure E.7: The tracks on Images (*bk*), (*bl*), (*bm*), (*bn*), (*bo*) and (*bp*) of the 64 images using original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19^{th}, 2007

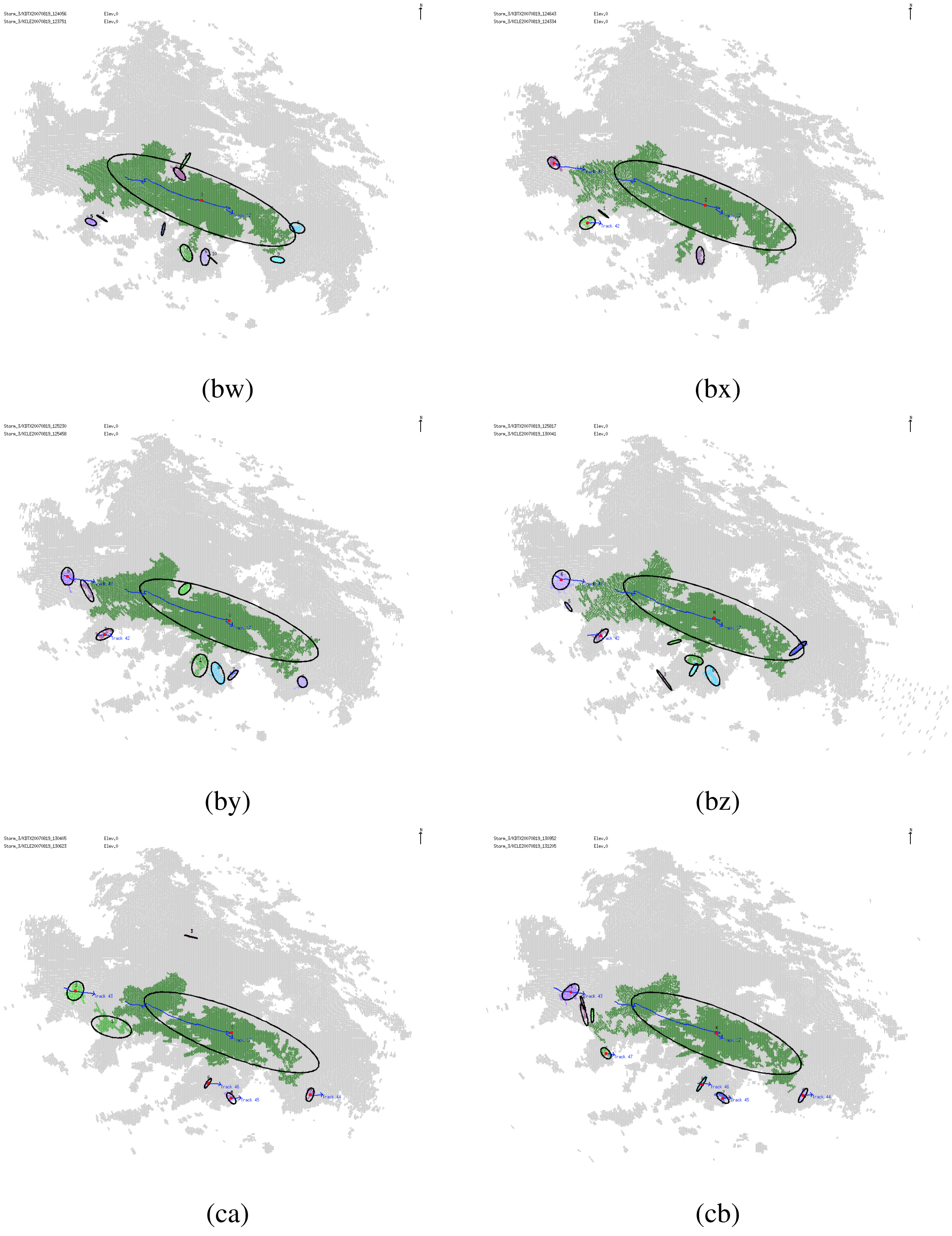


(bq) (br)

(bs) (bt)

(bu) (bv)

Figure E.8: The tracks on Images (*bq*), (*br*), (*bs*), (*bt*), (*bu*) and (*bv*) of the 64 images using original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19th, 2007

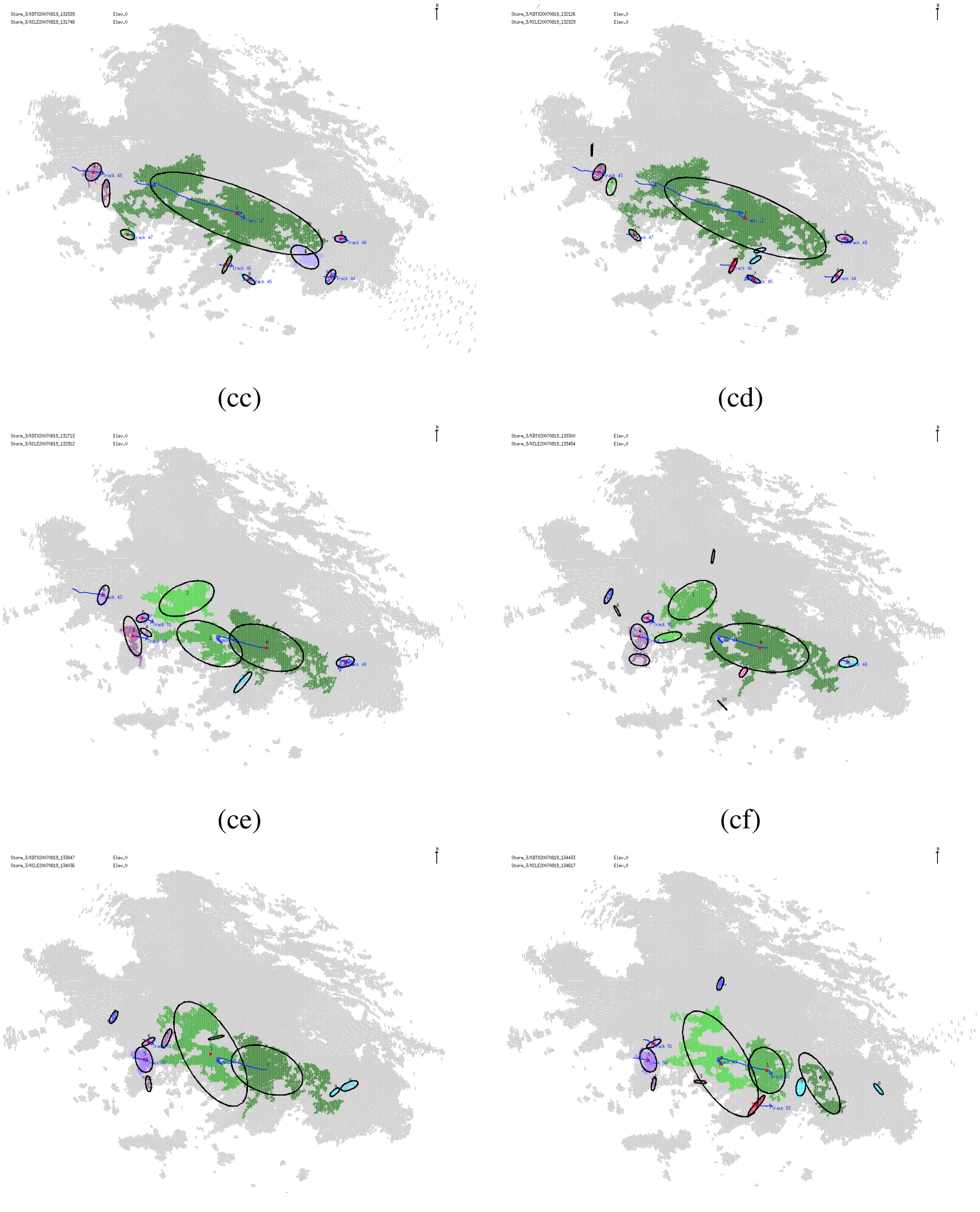


(bw) (bx)

(by) (bz)

(ca) (cb)

Figure E.9: The tracks on Images (*bw*), (*bx*), (*by*), (*bz*), (*ca*) and (*cb*) of the 64 images using original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19^{th}, 2007

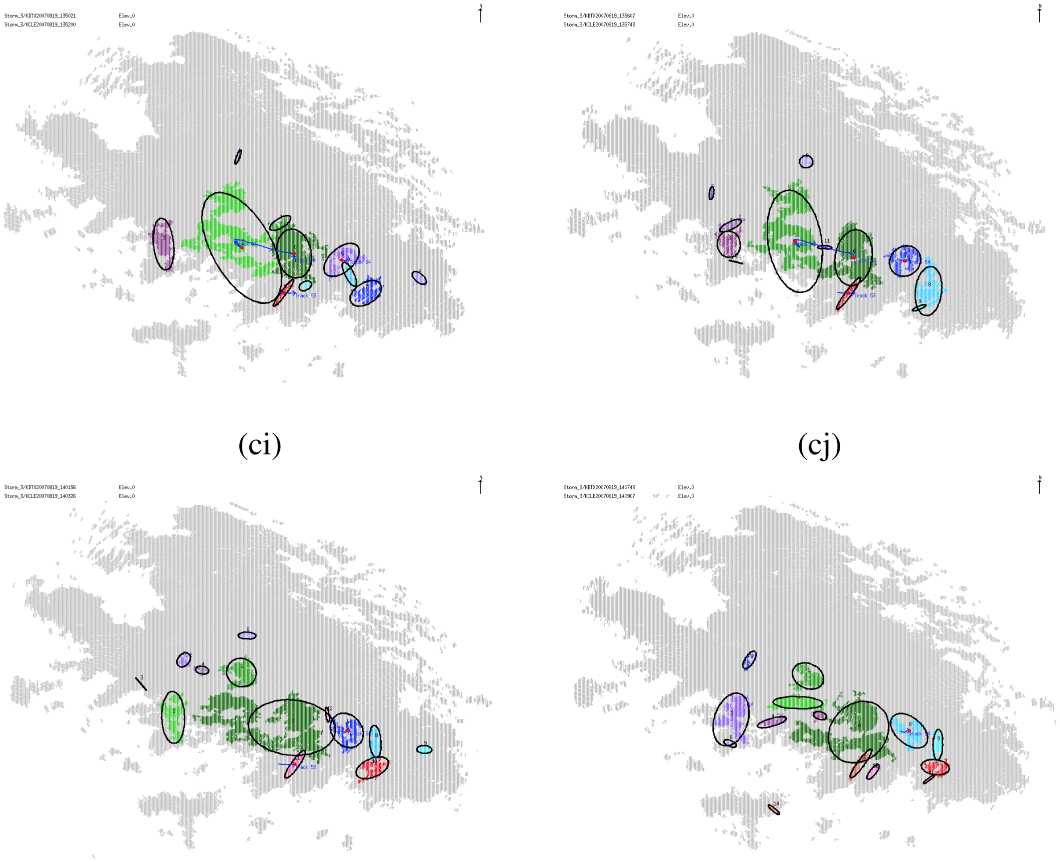


(cc) (cd)

(ce) (cf)

(cg) (ch)

Figure E.10: The tracks on Images (cc), (cd), (ce), (cf), (cg) and (ch) of the 64 images using original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19th, 2007



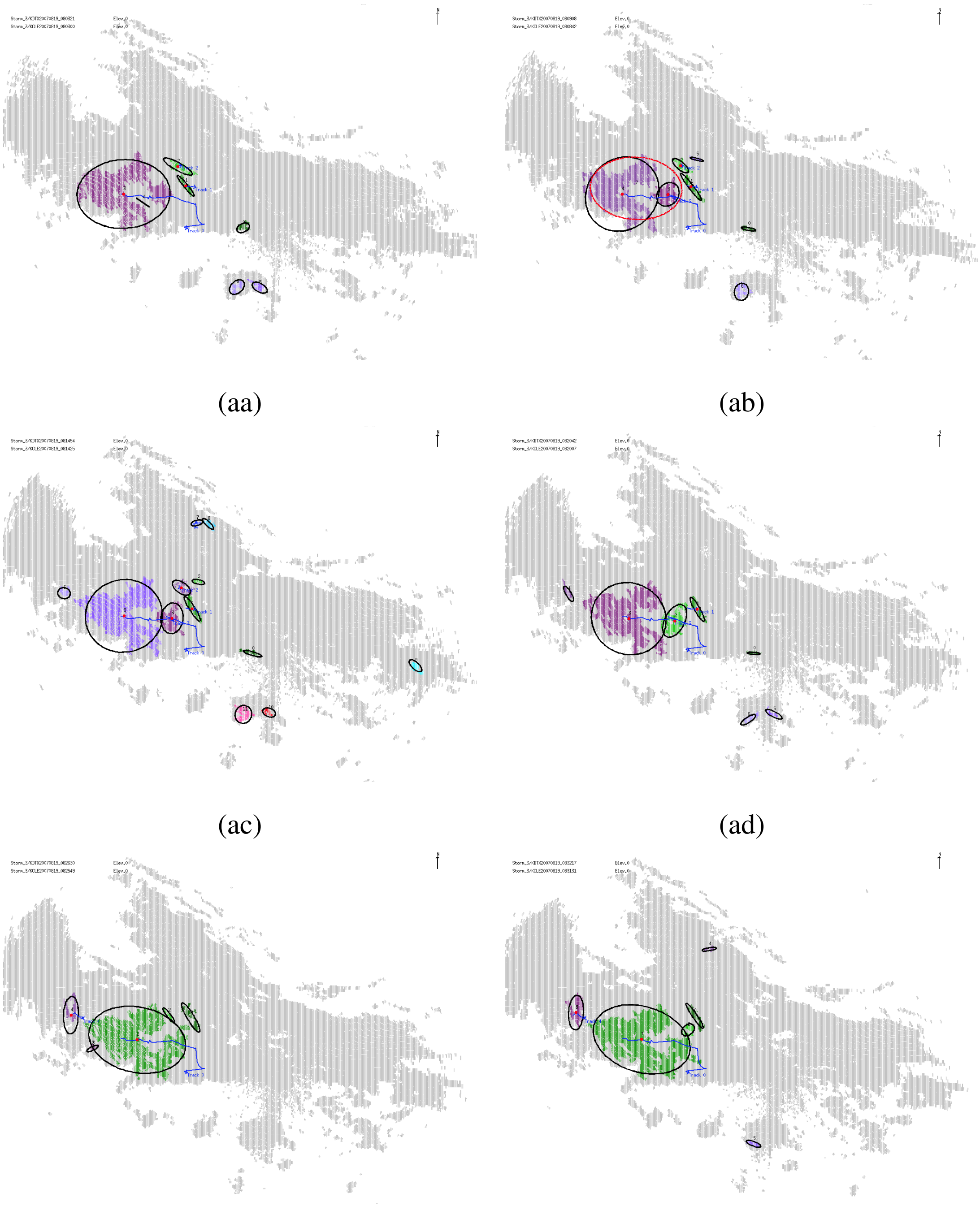
(ci) (cj)

(ck) (cl)

Figure E.11: The tracks on Images (ci), (cj), (ck) and (cl) of the 64 images using original relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19th, 2007

E.2 Image Sequence 2

The second group of figures displays the tracking results from the Detroit and Cleveland Doppler data on August 19th, 2007 using the advanced pseudo storm relaxation labeling algorithm.

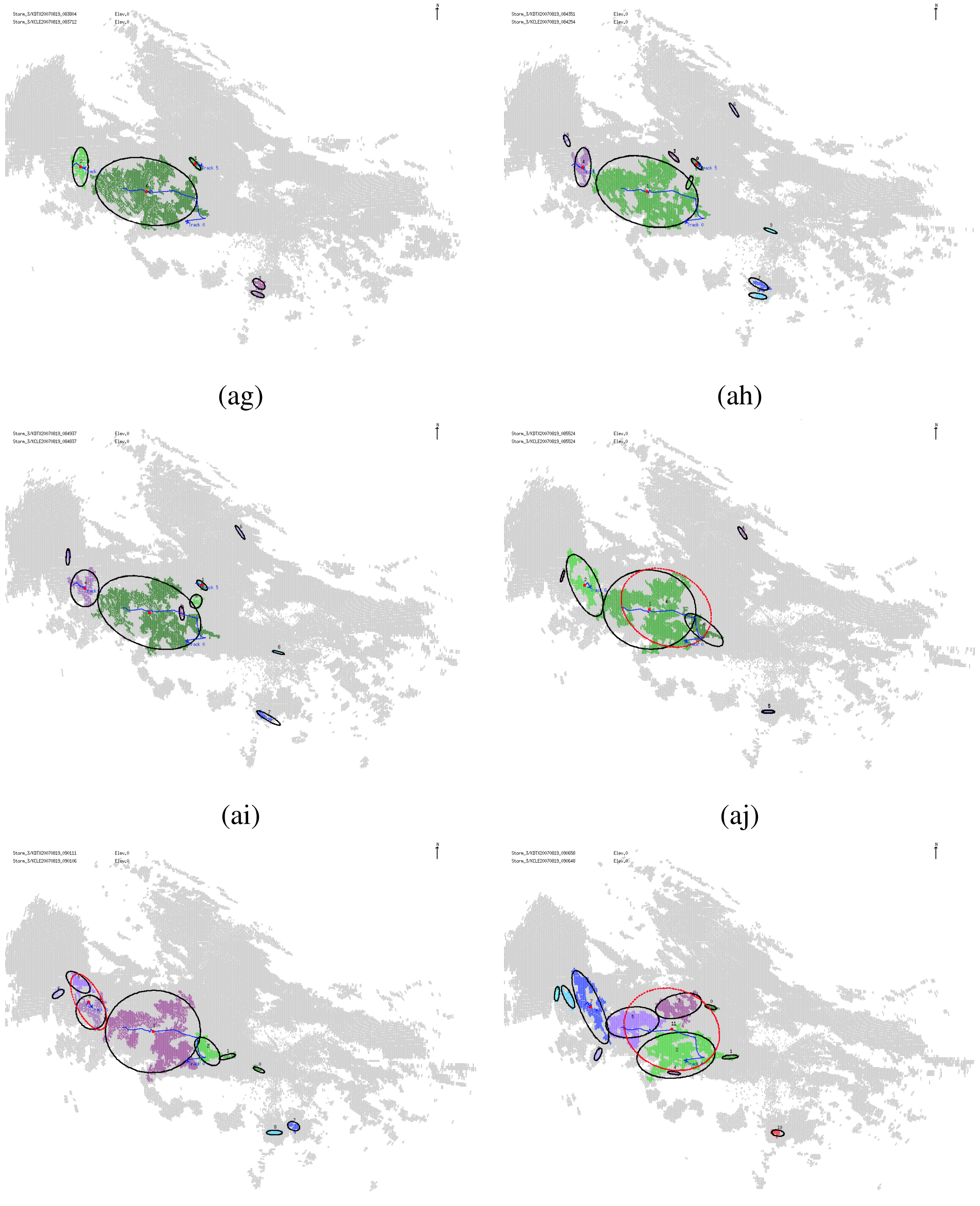


(aa) (ab)

(ac) (ad)

(ae) (af)

Figure E.12: The tracks on Images (aa), (ab), (ac), (ad), (ae) and (af) of the 64 images using advanced pseudo storm relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19^{th}, 2007

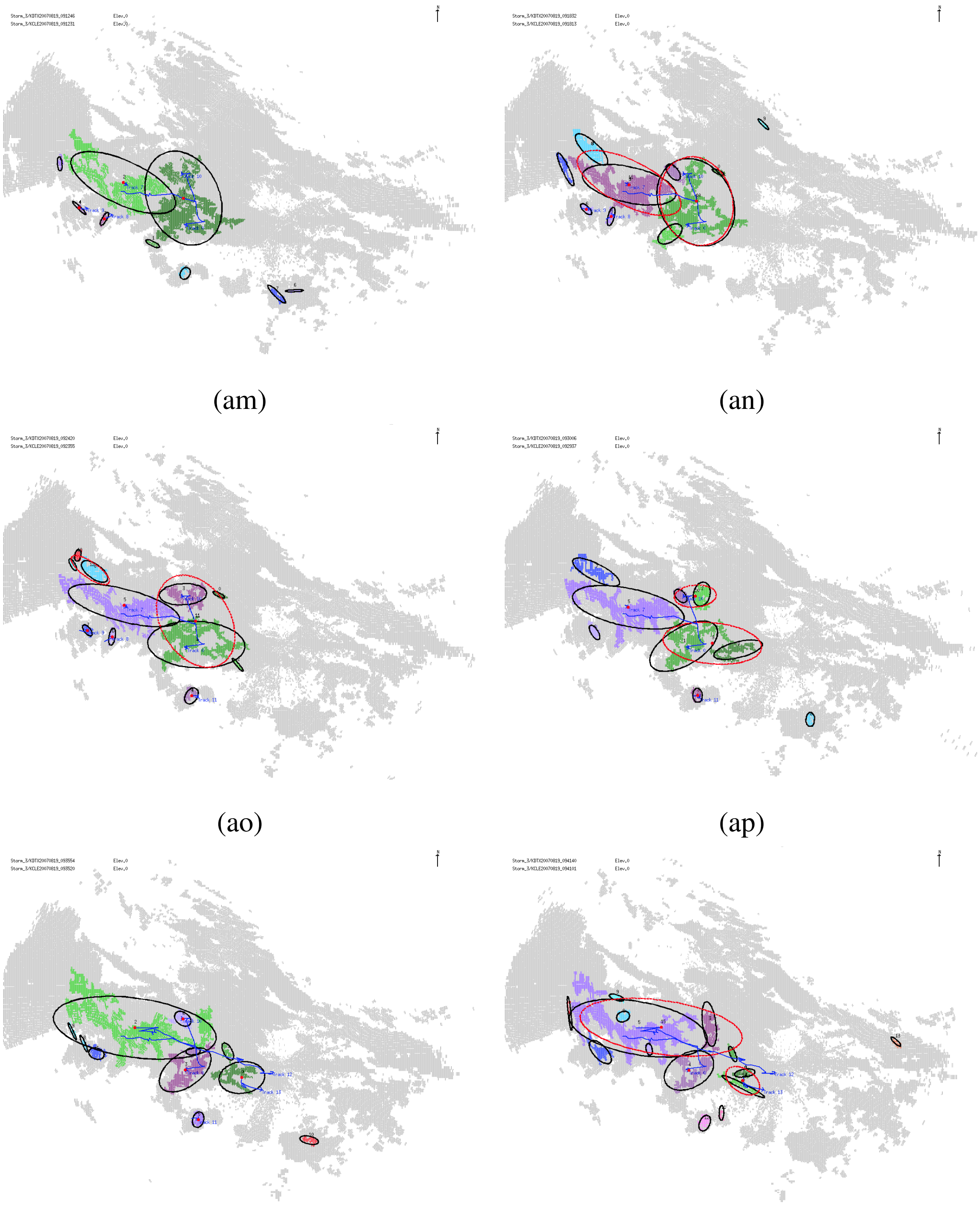


(ag) (ah)

(ai) (aj)

(ak) (al)

Figure E.13: The tracks on Images (*ag*), (*ah*), (*ai*), (*aj*), (*ak*) and (*al*) of the 64 images using advanced pseudo storm relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19th, 2007

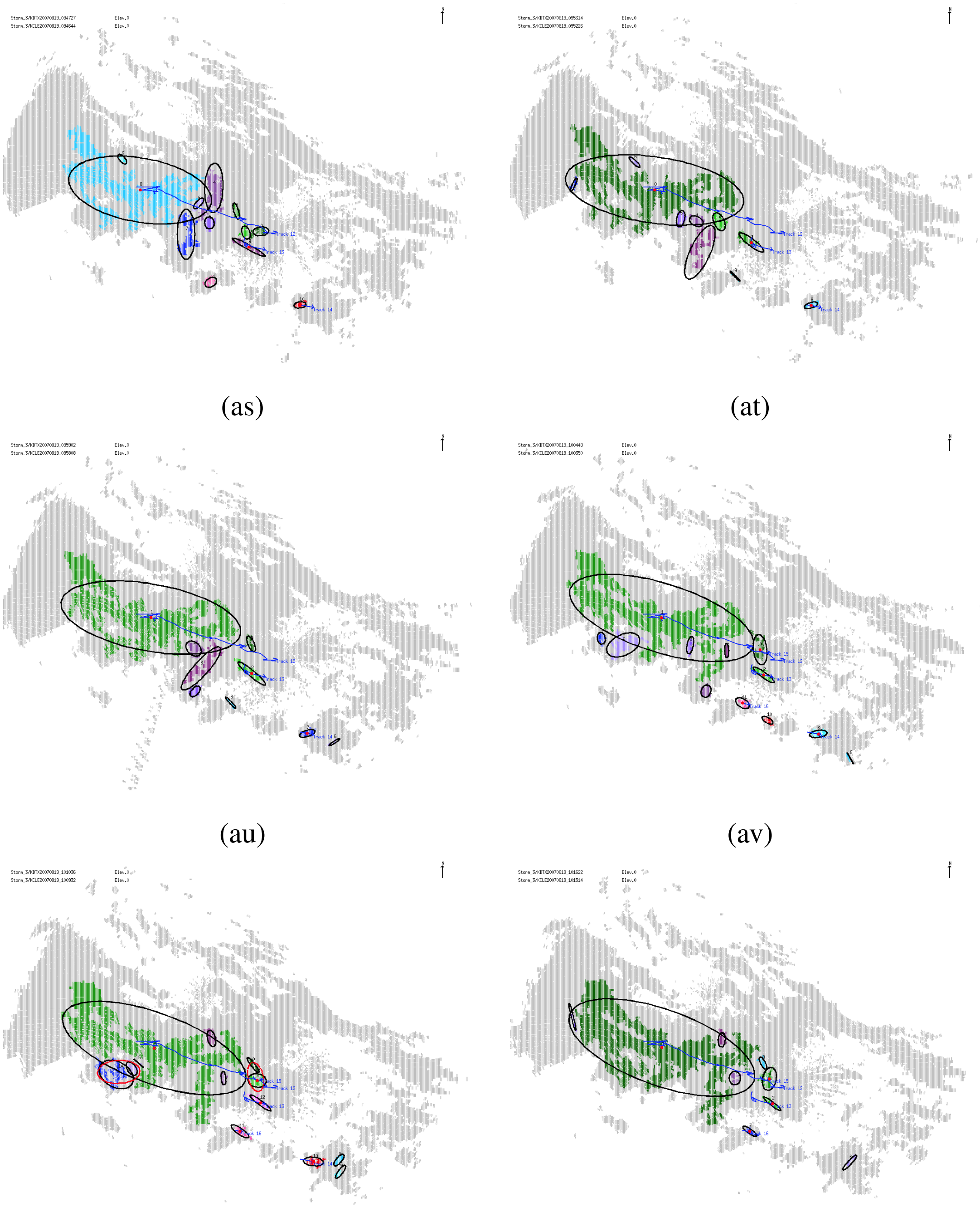


(am) (an)

(ao) (ap)

(aq) (ar)

Figure E.14: The tracks on Images (*am*), (*an*), (*ao*), (*ap*), (*aq*) and (*ar*) of the 64 images using advanced pseudo storm relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19th, 2007

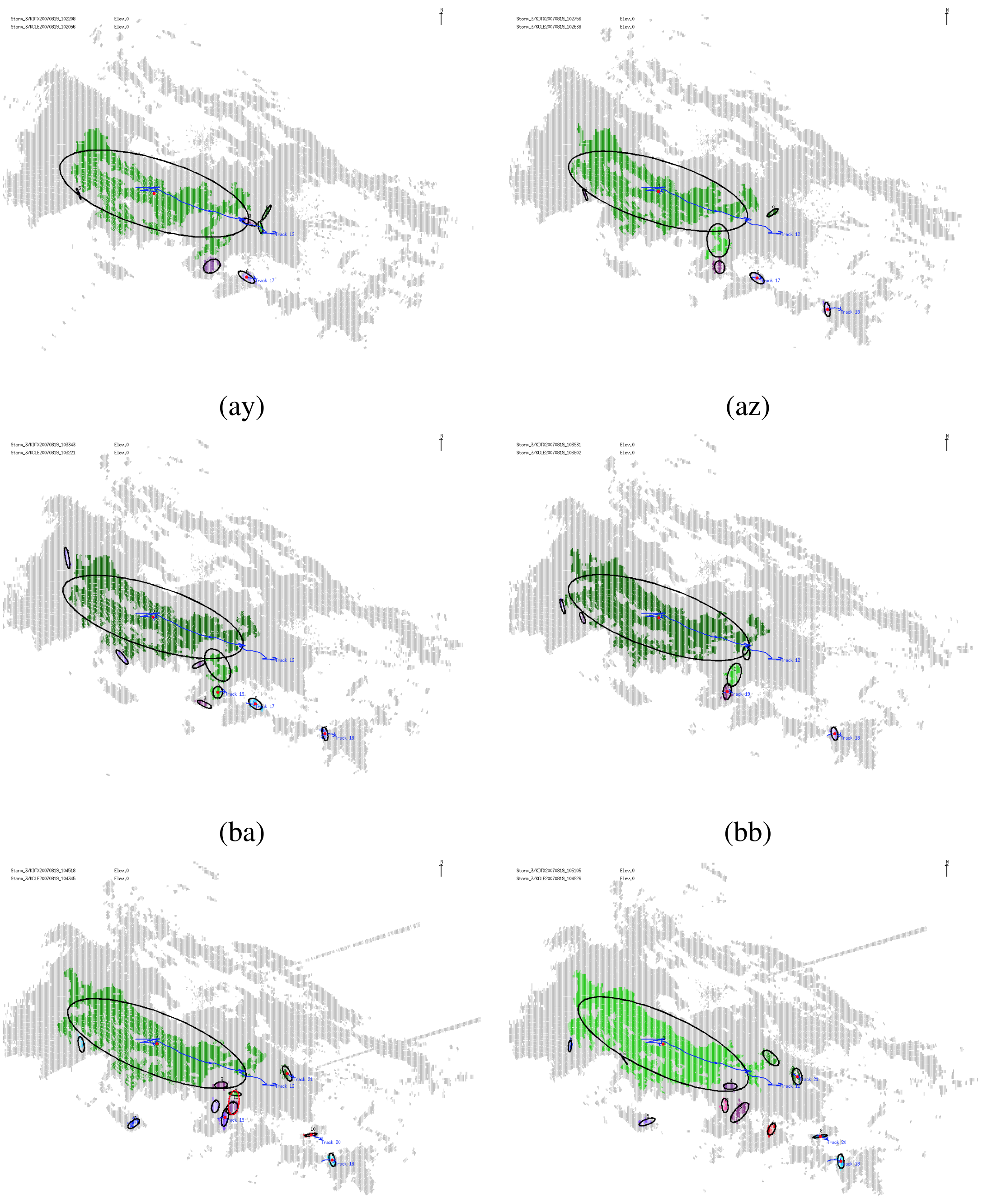


(as) (at)

(au) (av)

(aw) (ax)

Figure E.15: The tracks on Images (as), (at), (au), (av), (aw) and (ax) of the 64 images using advanced pseudo storm relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19^{th}, 2007

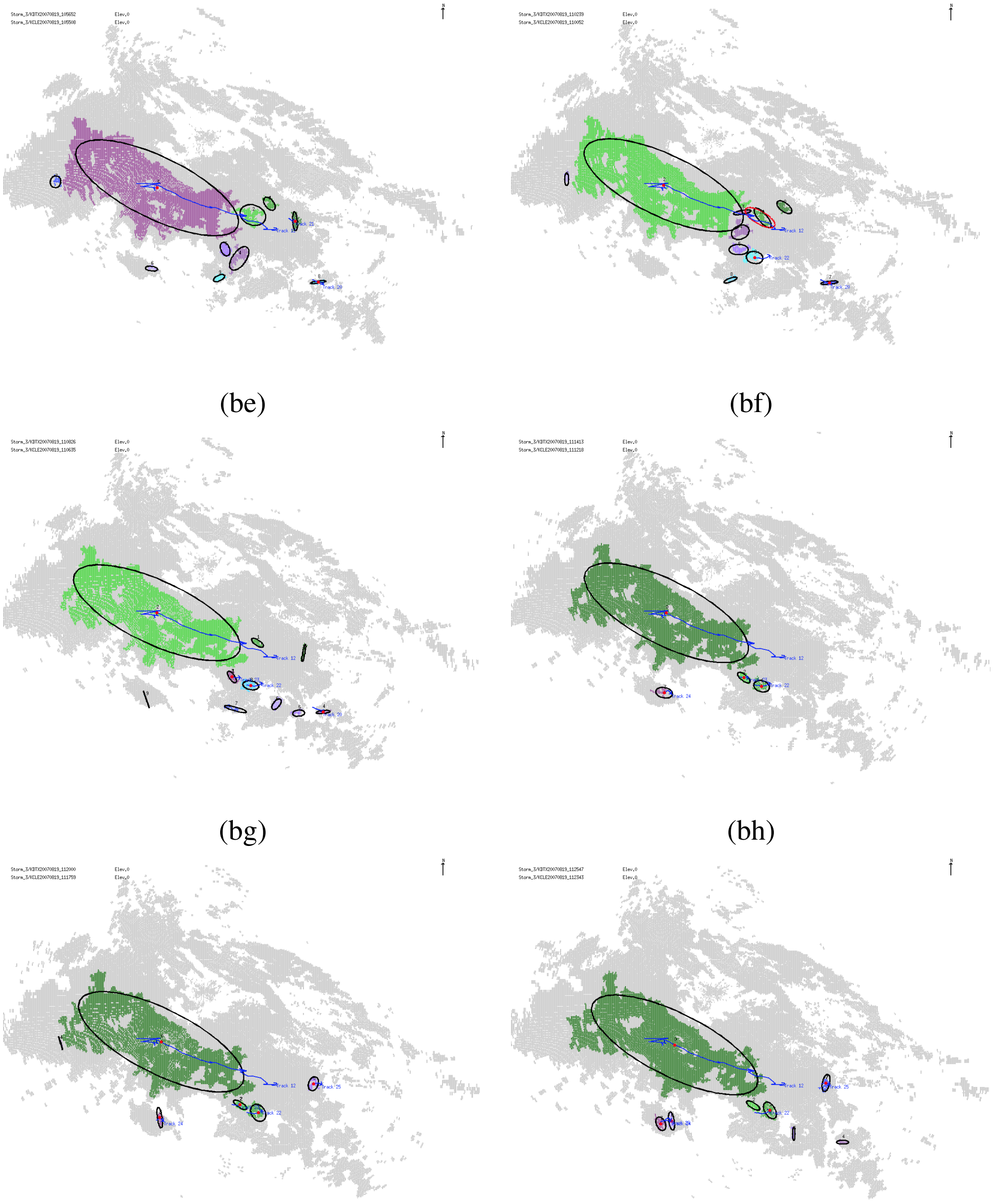


(ay) (az)

(ba) (bb)

(bc) (bd)

Figure E.16: The tracks on Images (*ay*), (*az*), (*ba*), (*bb*), (*bc*) and (*bd*) of the 64 images using advanced pseudo storm relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19th, 2007

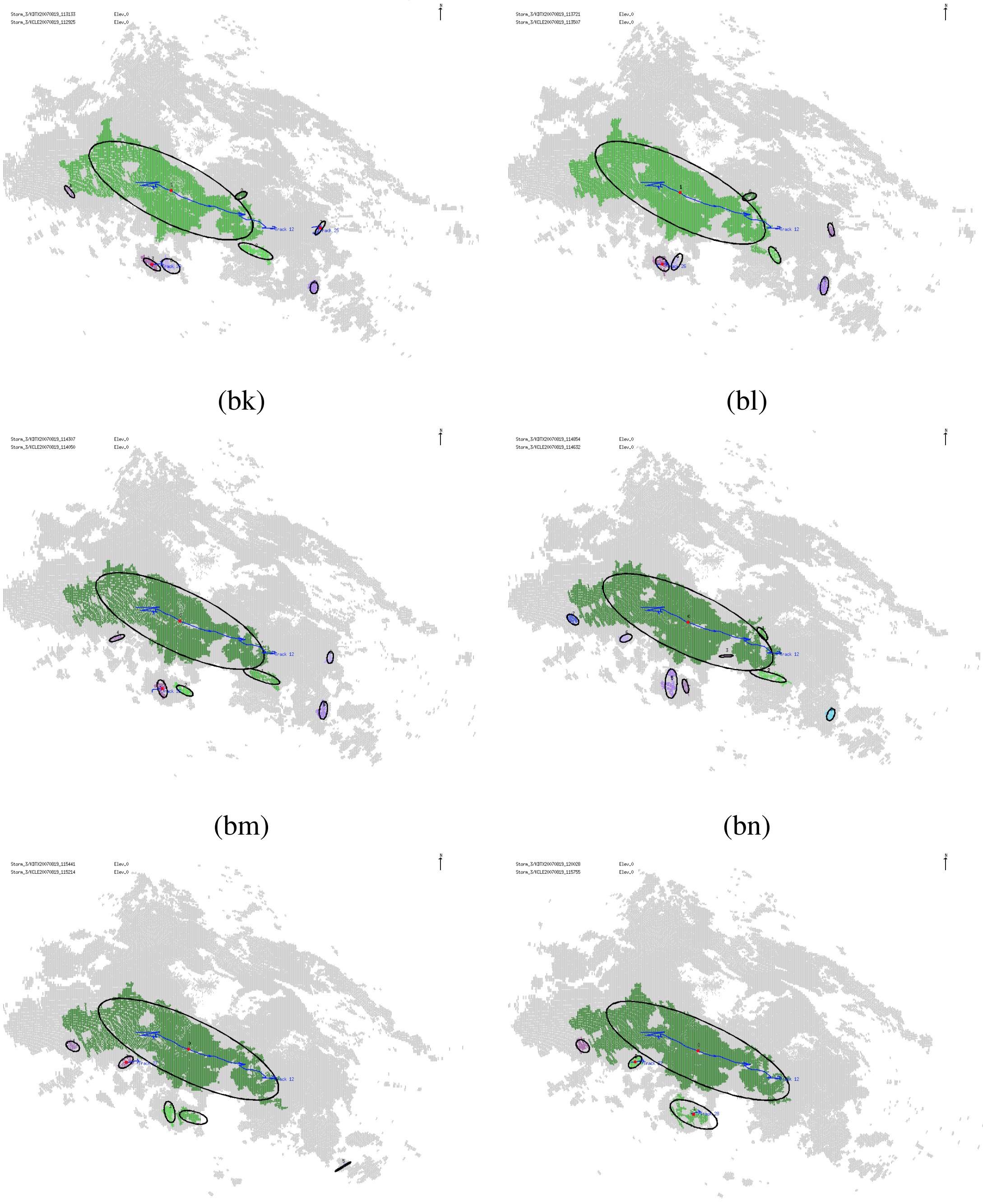


(be) (bf)

(bg) (bh)

(bi) (bj)

Figure E.17: The tracks on Images (be), (bf), (bg), (bh), (bi) and (bj) of the 64 images using advanced pseudo storm relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19th, 2007

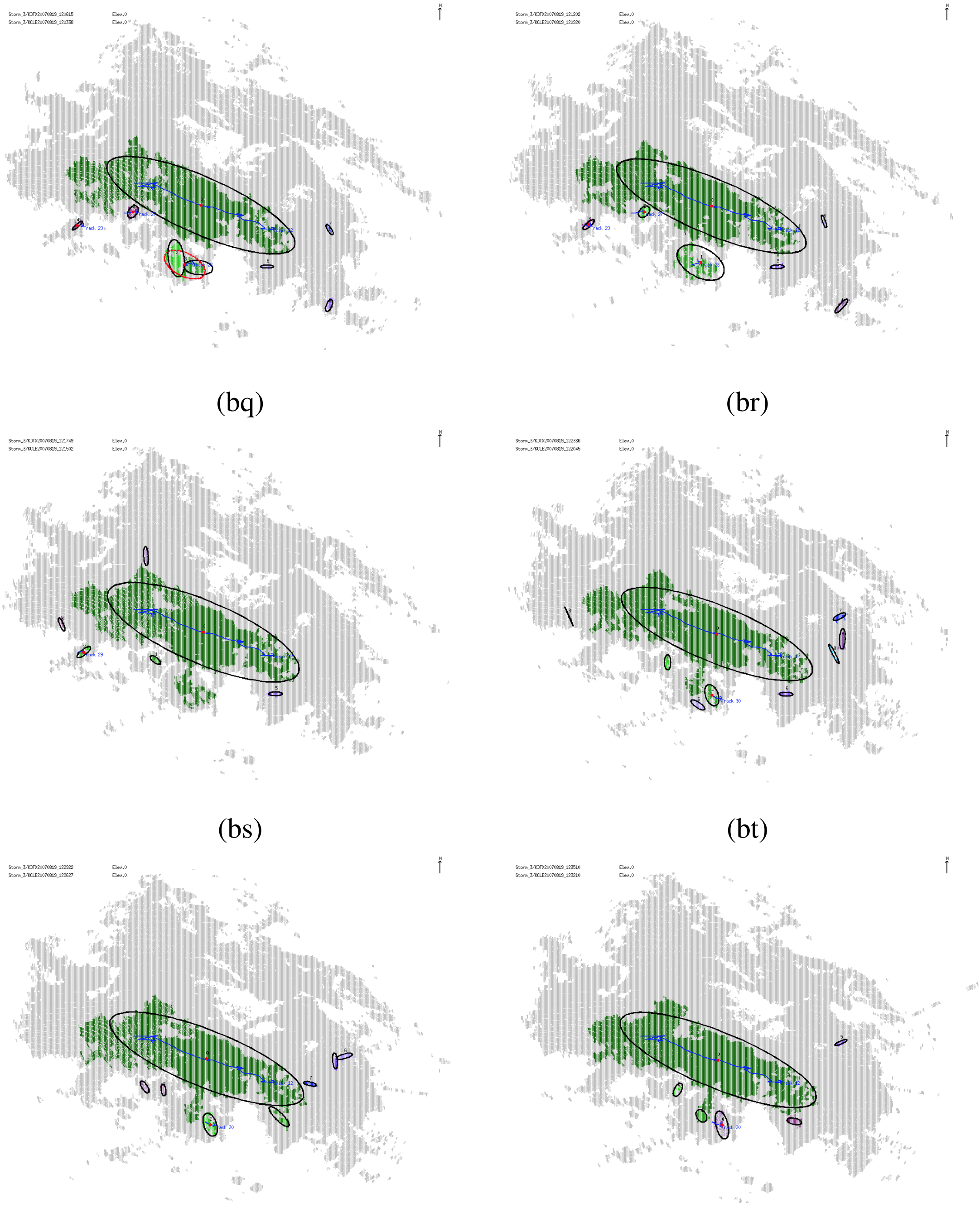


(bk) (bl)

(bm) (bn)

(bo) (bp)

Figure E.18: The tracks on Images (*bk*), (*bl*), (*bm*), (*bn*), (*bo*) and (*bp*) of the 64 images using advanced pseudo storm relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19th, 2007

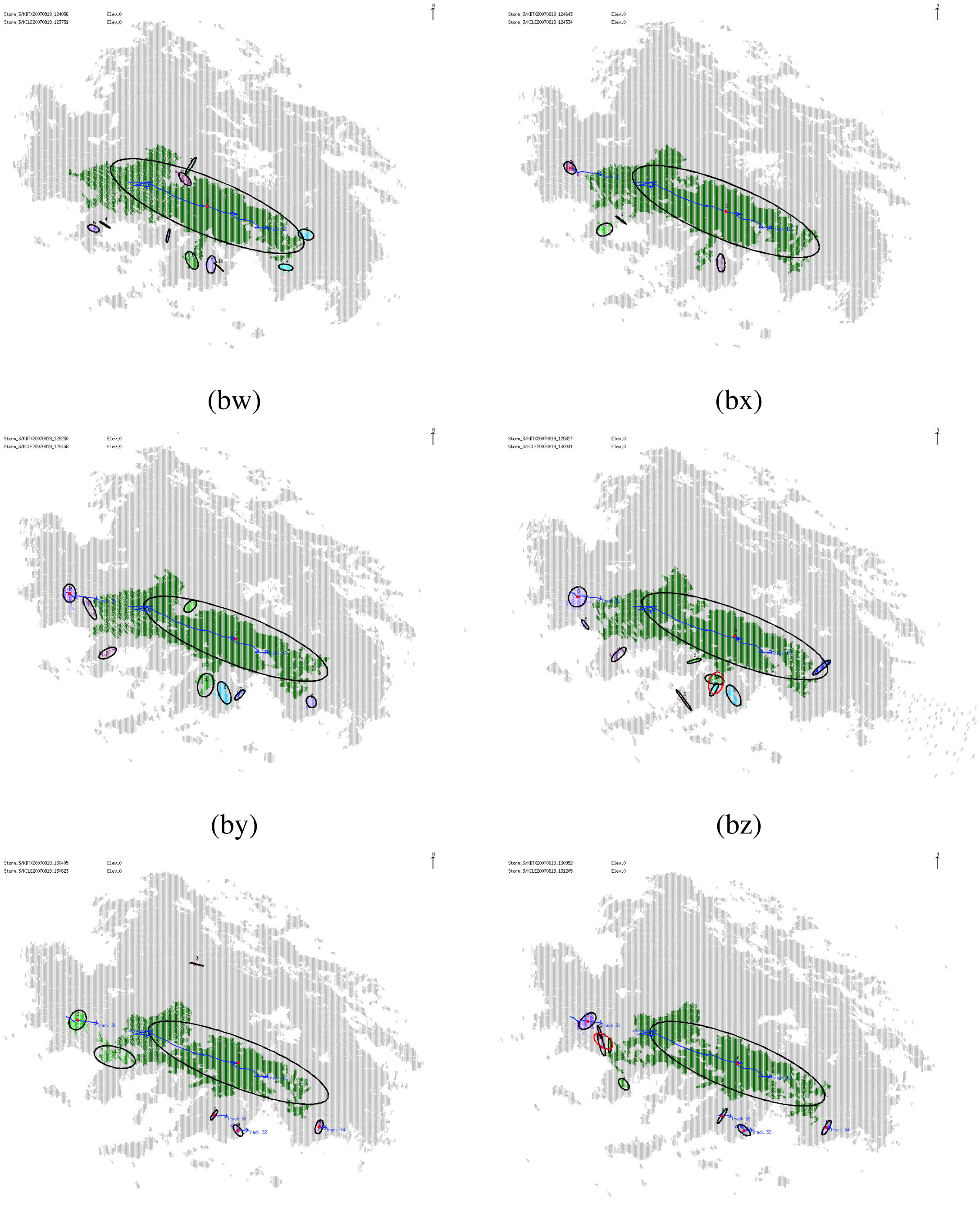


(bq) (br)

(bs) (bt)

(bu) (bv)

Figure E.19: The tracks on Images (bq), (br), (bs), (bt), (bu) and (bv) of the 64 images using advanced pseudo storm relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19th, 2007

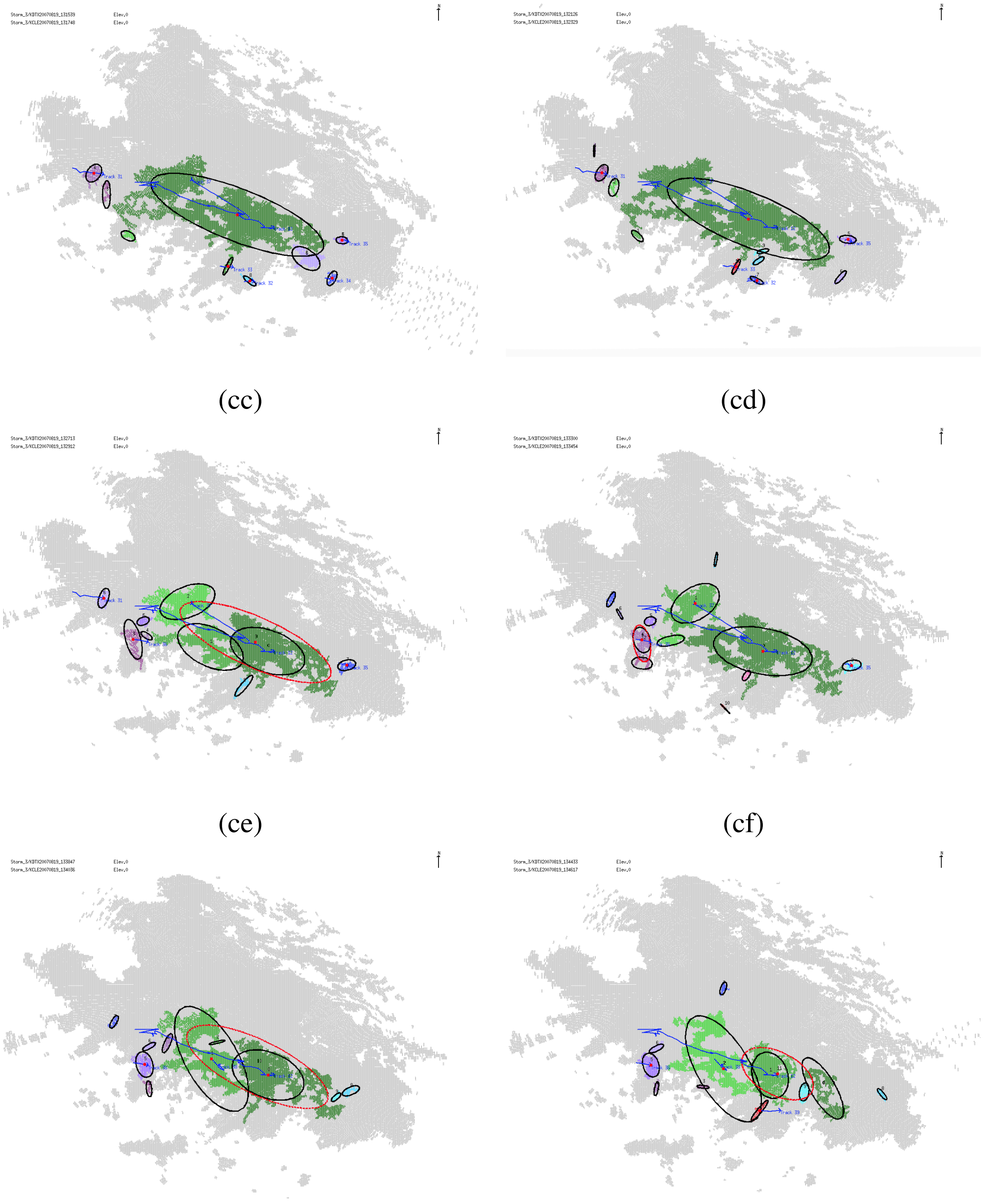


(bw) (bx)

(by) (bz)

(ca) (cb)

Figure E.20: The tracks on Images (*bw*), (*bx*), (*by*), (*bz*), (*ca*) and (*cb*) of the 64 images using advanced pseudo storm relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19th, 2007

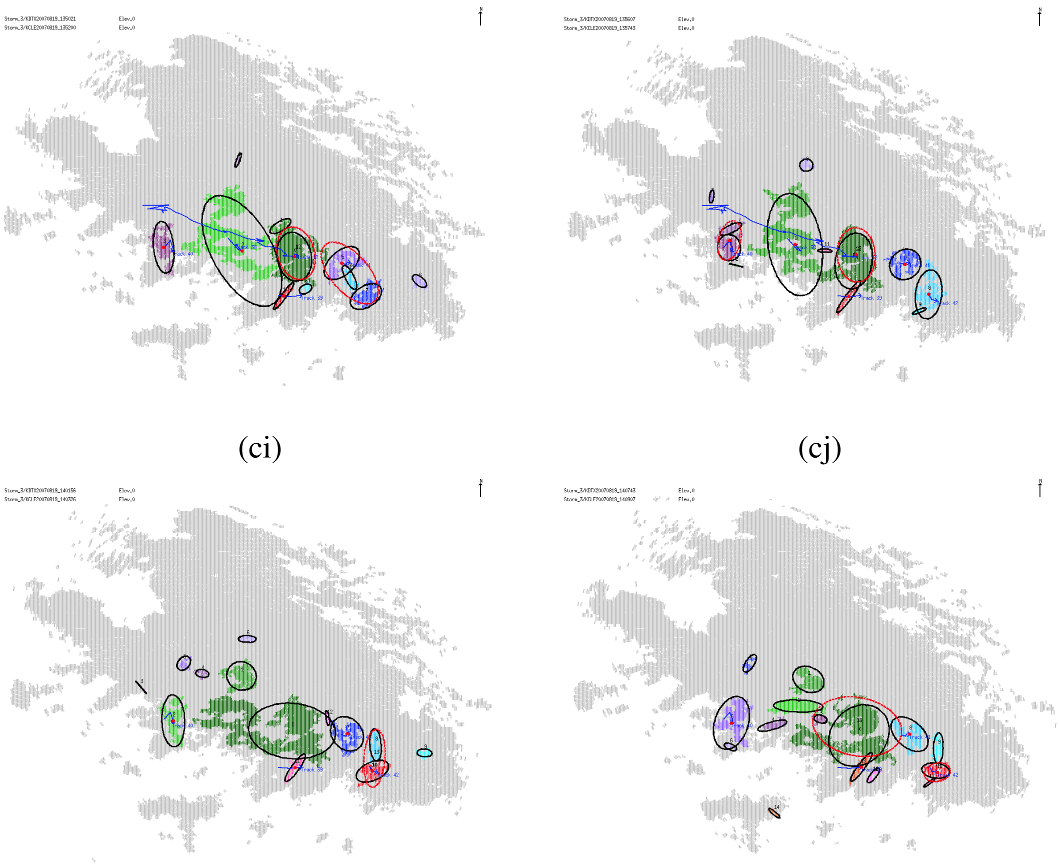


(cc) (cd)

(ce) (cf)

(cg) (ch)

Figure E.21: The tracks on Images (cc), (cd), (ce), (cf), (cg) and (ch) of the 64 images using advanced pseudo storm relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19th, 2007



(ci) (cj)

(ck) (cl)

Figure E.22: The tracks on Images (*ci*), (*cj*), (*ck*) and (*cl*) of the 64 images using advanced pseudo storm relaxation labeling algorithm from the Detroit/Cleveland Doppler data on August 19th, 2007

Curriculum Vitae

Name: Yong Zhang

Post-Secondary Education and Degrees:

PhD. Computer Science
The University of Western Ontario
London, Ontario, Canada
2006 - 2012

MSc. Computer Science
Trent University
Peterborough, ON
2004 - 2006

B.E. Computer Science and Technology
Shandong University
Jinan, P.R.China
1999 - 2003

Related Work Experience:

Software Developer
IBM Toronto Lab
2010 - 2011

Research Assistant
The University of Western Ontario
2008 - 2012

Teaching Assistant
The University of Western Ontario
2008 - 2012

Publications:

Yong Zhang, John Barron and Robert Mercer (2010),
“Refinement of 3D Doppler velocity using 3D windprofiler data”,
Twelfth International Workshop on Technical and Scientific
Aspects of MST Radar, (MST12), London, Ontario, May,
pp 183–186.

Yong Zhang, John Barron and Robert Mercer (2011),
“3D optical flow from single and dual Doppler radars”,
14th International Machine Vision and Image Processing
Conference, (IMVIP 2010), Limerick, Ireland, September,
pp 17–29.

Yong Zhang, John Barron, and Robert Mercer (2011),
“3D optical flow from Doppler and windprofiler radar data”,
Sixth International Conference on Computer Vision Theory and
Applications, (VISAPP 2011),Vilamoura, Portugal, Feburary,
pp 675–679.